Coupling neural networks to incomplete dynamical systems via variational data assimilation

by

Youmin Tang and William W. Hsieh

Oceanography/EOS

University of British Columbia, Vancouver, B.C., Canada V6T 1Z4

August 22, 2000

Submitted to Mon. Wea. Rev.

Abstract

The advent of the feed-forward neural network (NN) model opens the possibility of hybrid neural-dynamical models via variational data assimilation. Such a hybrid model may be used in situations where some variables, difficult to model dynamically, have sufficient data for modelling them empirically with an NN. We tested this idea of using NN to replace missing dynamical equations with the Lorenz (1963) 3-component nonlinear system, where we replaced one of the three Lorenz equations by an NN equation. In several experiments, the 4-D var assimilation approach is used to estimate: (1) the NN model parameters (26 parameters); (2) 2 dynamical parameters and 3 initial conditions for the hybrid model; and (3) the dynamical parameters, initial conditions and the NN parameters (28 parameters plus 3 initial conditions).

Two cases of the Lorenz model, (i) the weakly nonlinear case of quasi-periodic oscillations, and (ii) the highly nonlinear, chaotic case, were chosen to test the forecast skills of the hybrid model. Numerical experiments showed that for the weakly nonlinear case, the hybrid model can be very successful, with forecast skills similar to the original Lorenz model. For the highly nonlinear case, the hybrid model could produce reasonable predictions for at least one cycle of oscillation for most experiments, although poor results were obtained for some experiments. In these failed experiments, the data used for assimilation were often located on one wing of the Lorenz butterfly-shaped attractor, while the system moved to the second wing during the forecast period. The forecasts failed as the model had never been trained with data from the second wing.

1 Introduction

Numerical models have been widely used to simulate dynamical systems such as the atmosphere and the ocean. However, the physics in these models is usually incomplete, and some empirical approach is needed to patch up the missing physics. The first example is the inability of a numerical model, with its finite resolution, to represent sub-grid scale physical processes, thereby forcing numerical modellers to adopt parameterisation schemes for these processes. A second example arises from the fact that some variables of interest are not variables in the numerical model, e.g. precipitation and surface air temperature (which is generally not equivalent to the temperature at the lowest level of an atmospheric model). Often multiple linear regression (MLR) is used to empirically relate the model variables to the variables of interest, via schemes such as Perfect Prog and MOS (Model Output Statistics) (Wilks 1995). A third example arises in situations where replacing the physical equations by empirical ones results in large computational savings (and greater stability): In the equatorial Pacific, a simple empirical atmospheric model can be coupled to a dynamical ocean model, to form a hybrid coupled model. Here the wind stress is empirically estimated from the ocean variables either by a linear statistical method, such as MLR (Barnett et al. 1993) or singular value decomposition (SVD) (Syu and Neelin 1995), or by a neural network (Tang et al. 1999).

While it is an attractive idea to replace missing dynamical equations with empirical equations, there are serious technical problems– e.g. the dynamical equations are generally nonlinear, while the standard statistical methods are usually linear, hence not capable of simulating the evolution of nonlinear dynamical equations for extended periods. Neural networks (NN) have been known to be capable of simulating any nonlinear function, given a large enough network (Cybenko 1989). Recent advances in NN modelling have led to new techniques capable of nonlinearly generalizing the classical multivariate methods such as MLR, PCA (principal component analysis) and canonical correlation analysis (CCA) (Hsieh and Tang, 1998). As NNs originated from the field of artificial intelligence and robotics, it is tempting to ponder whether NN can benefit dynamical systems with incomplete physics the way robotic limbs have helped prosthetic. More specifically, could NN equations be used to simulate missing nonlinear dynamical equations? How to effectively couple an NN model to a dynamical model is a major challenge.

Variational data assimilation, especially via the adjoint approach, has become popular in assimilating data into numerical models (Daley 1991; Ghil and Malanotte-Rizzoli 1991; Bennett 1992; Navon 1998). The method is commonly used to optimally estimate model parameters or initial conditions, and is being implemented in operational weather and climate prediction models (e.g., Zhu and Navon 1999; Gauthier et al. 1999; Courtier et al. 1998; Ji et al. 1998). Hsieh and Tang (1998) noted that NN could be formulated as a variational assimilation problem, and suggested that NNs and dynamical models might be combined most naturally via a variational assimilation approach.

The objective of this paper is to show how an NN model can be coupled to a dynamical model (with incomplete physics) via variational assimilation. The chosen dynamical model is the simple Lorenz (1963) model, with 3 variables, arising from the Fourier truncation of the Rayleigh-Bénard equations describing atmospheric convection. In the field of data assimilation, the celebrated Lorenz model has served as a test bed for examining the properties of various data assimilation methods (Miller and Ghil 1990; Gauthier 1992; Miller et al. 1994; Evensen 1997) as the Lorenz model shares many common features with the atmospheric circulation and climate system in terms of variability and predictability (Palmer 1993). By adjusting the model parameters which control the nonlinearity of the system, the model can be used to simulate near-regular oscillations or highly nonlinear fluctuations.

This paper is structured as follows: Section 2 describes the hybrid Lorenz model, while section 3 shows some simple experiments involving the hybrid model. Section 4 gives a general formulation for coupling a NN to a dynamical model via variational data assimilation. Section 5 uses variation assimilation to estimate the NN parameters in the hybrid Lorenz model. Section 6 studies not only estimating the NN parameters, but also retrieving the dynamical model parameters and initial conditions.

2 Hybrid neural-dynamical Lorenz model

The non-dimensionalized Lorenz (1963) nonlinear system of 3 differential equations are

$$\frac{dX}{dt} = -aX + aY, \tag{1}$$

$$\frac{dY}{dt} = -XZ + bX - Y, \tag{2}$$

$$\frac{dZ}{dt} = XY - cZ. \tag{3}$$

where variables X, Y, and Z are related to the intensity of convective motion, and the temperature gradients in the horizontal and vertical directions respectively. The parameters a, b and c will be referred to as dynamical parameters, in contrast to the empirical parameters of the NN model.

The so-called observed data or true data are from the integration of the Lorenz equations (1)-(3) over 15,000 time steps at a step size h of 0.001, using a fourth-order Runge-Kutta integration scheme. As this system is very sensitive to changes in the initial conditions and model parameters, two cases are studied (Fig.1): The first case, called the weakly nonlinear case, with the parameters a, b and c set to 10, 28 and 8/3 respectively, and initial conditions for (X, Y, and Z) to (-9.42, -9.43, 28.3) (as in Gauthier 1992), displays near-regular oscillations with a gradually increasing amplitude in the devised integration period. The other case is the highly nonlinear case, with a, b and c set to 16.0, 120.1 and 4.0 respectively (as in Elsner and Tsonis 1993), and initial conditions to (22.8, 35.7, 114.9).

Next we assume the third Lorenz equation is unavailable, and we must approximate it empirically with an NN equation. Our hybrid model thus consists of:

$$\frac{dX}{dt} = -aX + aY, \tag{4}$$

$$\frac{dY}{dt} = -XZ + bX - Y, \tag{5}$$

$$\frac{dZ}{dt} = NN(X_t, Y_t, Z_t). \tag{6}$$

where NN is a feed-forward NN model (Hsieh and Tang 1998). X_t, Y_t and Z_t are the 3 input neurons (also denoted by v_i, i = 1,2,3), inputting the values of X, Y and Z at time t, and the single output neuron is dZ/dt. More details on the NN model are given in Appendix A.

3 Simple hybrid model experiments

A simple-minded approach to the missing third Lorenz equation, is to use data to get the NN eq.(6), and then integrate eqs.(4)-(6), with (6) as a replacement for the missing Lorenz equation. The NN has X, Y, and Z as the inputs and $(Z_{t+1} - Z_t)/h$ as the target during training (with the optimization minimizing the MSE (mean square error) between the target and the model output), so the model output is approximately dZ/dt. The optimization of the NN model was determined over a training period of 3000 time steps, and the following period of 1000 time steps was used to independently test the NN model skills in estimating dZ/dt. The appropriate number of neurons in the hidden layer is chosen based on the trade-off between under-fitting (too few neurons) and over-fitting (too many neurons) (Hsieh and Tang 1998). After trying models with different numbers of hidden neurons, we found that 5 hidden neurons yielded the best skills over the test period.

To alleviate the instability problems associated with NN modelling (Hsieh and Tang, 1998), an ensemble of 25 identical NNs were trained with random initial weights and biases, with the outputs from the 25 NNs used to provide an ensemble mean output. The skills of the ensemble mean generally exceeded the skills of an individual member (Hsieh and Tang, 1998). The ensemble mean NN simulations of dZ/dt for both the weakly nonlinear and the highly nonlinear cases during the training period and the test period are shown in Fig.2, where the NN gave good simulations of dZ/dt, although the oscillation peaks were underestimated.

Now that the third Lorenz equation seems to be reasonably approximated by the NN eq. (6), we proceed to integrate the hybrid system (4)-(6) from perfect initial conditions and compare the output with respect to that from the true Lorenz system (1)-(3). This simple-minded hybrid approach worked reasonably well in simulating the true Lorenz system for the weakly nonlinear case (Fig.3), but failed completely for the highly nonlinear case, where the hybrid model yielded X and Y values which were off by orders of magnitude. The highly nonlinear Lorenz system is known for its extremely high sensitivity to perturbations. Clearly, the NN approximation to the third Lorenz equation introduced significant errors in Z, which quickly caused the X and Y values to deviate far off course.

The defect of this simple approach lies in the fact that the optimization of the NN was achieved without imposing the dynamical constraints of (4) and (5). A better approach is variational assimilation, where the dynamical constraints are imposed.

4 The hybrid model using variational data assimilation

The scalar equations (4)-(6) can be expressed as a vector equation:

$$\frac{d\mathbf{v}}{dt} = \mathbf{f}(\mathbf{v}, \mathbf{p}, t) \tag{7}$$

where $\mathbf{v}$ is the vector denoting (X, Y, Z), and $\mathbf{p}$ is the parameter vector (which could contain the parameters of the NN or of the dynamical equations).

The variational assimilation method involves minimizing a quadratic cost function J subject to model constraint (7), where J is defined as

$$J(\mathbf{v}, \mathbf{p}, \mathbf{v_0}) = \int_0^T D(\mathbf{v}, t)dt \tag{8}$$

$$D = (\mathbf{v} - \mathbf{v_{obs}})^{tr}\mathbf{W}^{-1}(\mathbf{v} - \mathbf{v_{obs}}) \tag{9}$$

where the subscript obs denotes the observed values, the superscript tr the transpose, T the length of the assimilation window (also called training period sometimes), $\mathbf{v_0}$ the initial conditions, and $\mathbf{W}$ the covariance matrix of the measurement errors, assumed here to be diagonal.

Introduce the Lagrange function L,

$$L = J + \int_0^T \mathbf{v}^{*tr}[\frac{d\mathbf{v}}{dt} - \mathbf{f}]dt \tag{10}$$

where $\mathbf{v}^*(t)$ is a vector of Lagrange multipliers (or adjoint variables). After integration by parts, L becomes

$$L = [\mathbf{v}^{*tr}\mathbf{v}]_0^T + \int_0^T [D - (\frac{d\mathbf{v}^{*tr}}{dt}\mathbf{v} + \mathbf{v}^{*tr}\mathbf{f})]dt \tag{11}$$

The first order variation of L is

$$\delta L = [\mathbf{v}^{*tr}\delta\mathbf{v}]_0^T + \int_0^T [\nabla_\mathbf{v} D^{tr}\delta\mathbf{v} - [(\frac{d\mathbf{v}^*}{dt})^{tr}\delta\mathbf{v} + (\frac{\partial\mathbf{f}}{\partial\mathbf{v}})^{tr}\mathbf{v}^*\delta\mathbf{v} + (\frac{\partial\mathbf{f}}{\partial\mathbf{p}})^{tr}\mathbf{v}^*\delta\mathbf{p}]dt \tag{12}$$

The hybrid adjoint model can be obtained by letting $\delta L/\delta\mathbf{v} = 0$:

$$-\frac{d\mathbf{v}^*}{dt} = \frac{\partial\mathbf{f}}{\partial\mathbf{v}}\mathbf{v}^*(t) - \nabla_\mathbf{v} D, \tag{13}$$

$$\mathbf{v}^*(T) = 0. \tag{14}$$

According to (13) and (14), the formulas for computing the gradients of J with respect to $\mathbf{p}$ and $\mathbf{v}_0$ can be obtained by differentiating (12) with respect to these unknowns (Lu and Hsieh 1998):

$$\frac{\delta J}{\delta\mathbf{p}} = -\int_0^T (\frac{\partial\mathbf{f}}{\partial\mathbf{p}})^{tr}\mathbf{v}^* dt, \tag{15}$$

$$\frac{\delta J}{\delta\mathbf{v}_0} = -\mathbf{v}^*(0). \tag{16}$$

Details of the variational assimilation are given in Appendix B.

As the above assimilation process, also referred to as the training process, is completely subject to the model (4)-(6) after the initial guesses are given, it is called the strong continuity constraint assimilation scheme (SC).

5 Determining NN parameters via variational assimilation

In this section, we determine the NN parameters, i.e., the weights and biases, in the hybrid model by variational data assimilation. An NN with 5 hidden neurons is used. Preliminary experiments are done in Section 5a and 5b to determine the best first guesses and assimilation window sizes, respectively, before the main experiments are run in Section 5c. In this section, all experiments were performed with perfect initial conditions and dynamical parameters a and b.

a The initial guesses

Whether an assimilation process is successful depends greatly on the choice of the first guesses. To help getting good initial guesses, two assimilation schemes are introduced in addition to the SC scheme. Under SC, at each step of the integration, the initial conditions are the model outputs from the previous step (Fig. 4a). In the no continuity constraint assimilation scheme (NC), at each integration step, the initial conditions are the observed data, rather than the model output from the previous step (Fig. 4b).

Between these two extremes, we can have the partially strong continuity constraint assimilation scheme (PSC) (Fig.4c), where the assimilation window T is divided into smaller assimilation segments (AS). Within each AS, integration at each step uses the model output from the previous step as the initial conditions. Only at the start of the AS, are the data used directly as the initial conditions. Both SC and NC can be regarded as special cases of PSC, where the AS is T in SC and 1 time step in NC.

The reason for introducing NC and PSC is that using SC without good initial guesses generally does not lead to successful assimilation over a long assimilation window. Chopping the window into shorter AS means that the dynamic model constraints are no longer imposed over a long period, thereby making the nonlinear optimization a much easier task. Our strategy is to divide the assimilation process into 2 stages: (a) use less demanding schemes such as NC and PSC to obtain reasonable parameters estimates, which will then be used as initial guesses in (b) the full variational assimilation under SC.

For the weakly nonlinear case, the NC assimilation scheme was first applied to the hybrid model (4)-(6), with initial guesses for the NN parameters taken randomly from one of the 25 NNs in Section 3. The assimilation window is 3000 time steps. The NC scheme can produce good results during the training period but poor skill during the test period (Fig. 5). Since the AS is very short (1 time step) in the NC scheme, the initial values at each integration step vastly outweighs the importance of the dynamical constraints, hence the poor skill during the test period. However the result from the NC can provide reasonable first-guesses for the following PSC.

A series of PSC assimilations with AS = 100, 200, 500 and 1000 time steps was further performed with the hybrid model (in the weakly nonlinear case) to gradually improve the first guesses to be used in the stage (b) SC assimilation. A longer AS increases the importance of dynamical constraints relative to initial conditions in influencing the model outputs. The first guesses of the NN parameters in a PSC experiment are the parameters obtained from the previous PSC experiment with the shorter AS. The first PSC experiment (not shown) with AS = 100 time steps uses the results from the NC case (i.e. AS = 1 time step) for its first guesses.

The PSC assimilation over the training period, i.e. a window of 3000 time steps, with AS = 200 time steps (Fig.6), and AS = 1000 time steps (Fig.7), reveal that the test skills improved with the increase of the AS. The parameters estimated from the PSC experiment with AS= 1000 time steps will serve as the first guess for the stage (b) SC assimilation (in the next Section) for the weakly nonlinear case, where the assimilation window is also taken to be 1000 time steps.

This approach of using a series of PSC assimilations to improve on the parameter estimates to be used as initial guesses in the stage (b) SC assimilation did not work for the highly nonlinear case. Unlike the weakly nonlinear case, the first guesses from PSC, if found, is only effective for a specific experimental configuration (e.g. AS, window length, etc.). Hence, for the highly nonlinear case, the parameters estimated from the NC assimilation over 3000 time steps are directly used as the first guesses for the stage (b) SC assimilation.

b Impact of assimilation window in the highly nonlinear case

The greatest difficulty in variational assimilation, namely the presence of multiple minima in the cost function arising from the nonlinearity of the problem, often renders the problem intractable.

The number of local minima dramatically increases with the window T for a strongly nonlinear dynamical system. Gauthier (1992) and Miller et al (1994) have shown with the Lorenz model that variational assimilation is effective only for sufficiently short windows. In the weakly nonlinear case (Section 5a), T did not have a major impact; hence, the following discussion will only focus on the highly nonlinear case.

To study the impact of the window, SC assimilation experiments were performed, where T was varied between 100, 300, 500 and 800 time steps. For each T, 10 experiments were performed with the same first guesses, but the assimilation periods were shifted by 100 time steps between one experiment and the next.

The effect of T on the assimilation is different for training period (Fig.8a) and the test period (Fig.8b). With a short window ($T = 100$ time steps), the correlation skills were excellent for all 3 variables during training, but yielded no prediction skills over the following test period of 300 time steps. As T increased to 300 time steps and then 500 time steps, there were increases in the prediction skills (Fig.8b), but as T increased to 800, the prediction skill dropped sharply, as the problem with local minima in the cost function worsened with increasing T. Hence in the following SC assimilation experiments, the window will be 500 time steps for the highly nonlinear case, and 1000 time steps for the weakly nonlinear case. Following the training (assimilation) period, there is a test period of 300 time steps for the highly nonlinear case, and 1000 time steps for the weakly nonlinear case.

c SC assimilation

We now perform the main SC assimilation experiments, repeated 100 times, to examine the skills of the hybrid model for the weakly nonlinear and highly nonlinear cases. The choices of windows and first guesses are those found in Sections 5a and 5b. The 100 experiments are implemented with their assimilation periods shifted by 100 time steps from one experiment to the other.

For the weakly nonlinear case, the average correlation between model and data (over 100 experiments) exceeds 0.96 for each of X, Y and Z during both the training period and the test period (Fig.9a). The relative estimate error, REE, i.e. $\sum$ (estimated value-true value)^2/ $\sum$ (true value)^2) is very small, less than 0.004 (Fig.9b). Thus, via variational assimilation, a hybrid model has successfully reconstructed the Lorenz model in the weakly nonlinear case. For comparison, the results of the simple hybrid model without data assimilation from Section 3 are also show in Fig.9. The improvement due to variational data assimilation is dramatic for X and Y.

In contrast, the highly nonlinear case has much lower skill. During training, the skills were good– 80% of the experiments had correlation above 0.75, and 60% had REE under 0.05 (not shown). However, during the test period, the correlation skill and REE attained were much poorer (Fig.10).

Generally, for the highly nonlinear case, there were three classes of assimilation results found in the 100 experiments. A typical example of the class 1 results is shown in Fig. 11, showing successful assimilation. About 15% of the total experiments is of this class. Almost 60% of the total experiments belongs to a class 2 of moderately successful assimilation, as illustrated by the example in Fig.12, where during the test period, the forecast skills were fairly good for the first 200 time steps. The remaining 25-30% of the experiments belonged to a class 3 of failed forecasting, as illustrated in Fig.13, where despite excellent fitting during the training period, the forecast results were totally wrong.

To see why there is a drastic difference between class 1 and class 3 behaviour, we plotted the system trajectories of the two cases in the $X-Y-Z$ space (Fig.14). The Lorenz attractor for the highly nonlinear case is known to have a 'butterfly' shape with two wings. In the class 1 experiment, both wings of the Lorenz attractor were covered by the training data trajectory. In contrast, with the class 3 example, the training data trajectory resided in one wing. The system then evolved to the other wing during the testing period. As the model had never been trained with data from the second wing, it obviously could not forecast properly in that regime, hence the disastrously poor forecast results found in Fig.13. Thus the bimodality of the highly nonlinear Lorenz system created cases where the training data covered only one wing of the attractor, and the model thus trained had no capability of forecasting the transition to the other wing.

One would be tempted to think that the problem could be corrected by simply using longer training periods, so that the full Lorenz attractor would have been learned by the model. Unfortunately, a longer training period greatly increases the difficulty in the nonlinear optimization due to multiple local minima in the cost function, leading eventually to the failure of the variational assimilation method.

6 Determining dynamical parameters and initial conditions by the hybrid model

In variational data assimilation, parameters and initial conditions can be jointly estimated (e.g., Le Dimet and Talagrand 1986; Tziperman et al 1989; Lu and Hsieh 1998; Zhu and Navon, 1999). We next conducted an experiment, for the weakly nonlinear case, to estimate simultaneously the dynamical parameters a and b, and the initial conditions of X, Y and Z in the hybrid Lorenz system (4)-(6). The NN from Section 5a (i.e. PSC assimilation with AS = 1000 time steps) was used.

For the initial guesses, a and b and the initial conditions were all scaled down by 10% from their true values. The result of retrieving the 2 parameters and 3 initial conditions by the hybrid model with an assimilation period of 3000 steps under SC showed the retrieval to be very good, as the REEs were all of the order of 10^{-3} - 10^{-4} (not shown). Next the true values were scaled down by 20% to provide the initial guesses, and the retrieved results were still generally good (Fig. 15). But when the initial values were the true values scaled down by 30%, the retrieval nearly failed (not shown).

Next, we had the NN parameters retrieved as well during the assimilation, i.e. we conducted 100 additional SC experiments to estimate simultaneously the NN parameters, the dynamical parameters a and b, and the initial conditions of X, Y and Z, for the weakly nonlinear case. The first guesses for the initial conditions and the parameters a and b were simply the true values scaled down by 10%. The choice of the first guesses of the NN model parameters, cost function and assimilation window were the same as those discussed in the previous section.

The average correlation skills over the 100 experiments are shown in Fig.16, while the average REE are all around 0.05 (not shown). The correlation and REE were generally poorer than those in Fig.9, where only the NN parameters were retrieved. This agrees with many studies which found that the assimilation quality deteriorated while retrieving more model unknowns (Thacker and Long 1988; Lu and Hsieh 1998), as the problem of multiple minima in the cost function could become worse when retrieving more model unknowns.

Likewise, 100 experiments for the highly nonlinear case were also carried out to simultaneously estimate the NN parameters, the dynamical parameters a and b and the initial conditions. The first guesses were the true values scaled down by 10%, and the first guesses of the NN model, the cost function and assimilation window were still taken from Section 5. While the skills in the training period were as good as those retrieving the NN parameters only (not shown), the skills during the testing period (Fig.17) were considerably lower than those found when only NN parameters were retrieved (Fig.10). Class 1 examples like Fig. 11 can hardly be found now.

7 Summary and discussion

In this study, a hybrid neural-dynamical variational data assimilation procedure aimed at simulating missing dynamical equations was formulated. This procedure was then applied to the Lorenz model, where the third dynamical equation of the model was missing and had to be simulated empirically by an NN equation.

First guesses in variational data assimilation can be vital to retrieving the model parameters and initial conditions successfully. In order to get reasonable first guesses, the NC (No Continuity constraint) and PSC (Partial Strong Continuity constraint) schemes were proposed in this study. Experiments show that such a treatment of choosing first guesses is effective in the weakly nonlinear case, which allows the cost function to be successfully optimized in almost all experiments under SC (Strong Continuity constraint). For the highly nonlinear case, the NC scheme provides reasonable first guesses for the SC experiments.

The length of the assimilation window is very significant for the highly nonlinear case. When the window is either too short or too long, the prediction skills are poor. This strict demand for an appropriate assimilating window results from a delicate balance between the need for more data (longer window) and the avoidance of severe multiple minima during optimization with the highly nonlinear model, which calls for a short window.

Our numerical experiments showed that the hybrid model based on NN and variational assimilation is successful in simulating the weakly nonlinear Lorenz model, with forecast skills similar to the original Lorenz model. For a highly nonlinear Lorenz system, the hybrid model can also produce reasonable skills of simulations and predictions for most experiments, although the assimilation basically failed for a fair portion of the experiments. Deterministic optimization algorithms often have difficulties reaching the global minimum in the cost function, due to the presence of local minima. It has been shown that the variational assimilation with deterministic optimization algorithms often loses its power for strongly nonlinear models (Miller et al. 1994; Evensen 1997). Stochastic optimization methods such as simulated annealing may help (Kruger 1993; Hannachi and Legras 1995). However, even if the global minimum is found, the data in the assimilation window might still only consist of data from one wing of the butterfly-shaped attractor, whereas the data in the forecasting test period could lie in the second wing, thereby resulting in failed forecasts. Alternatively, other assimilation approaches, e.g. the Extended Kalman filter could be more powerful in dealing with strongly nonlinear systems (Miller et al. 1994; Evensen 1997; Houtekamer and Mitchell, 1998). The possibility of a NN coupled with a dynamical model with an Extended Kalman filter is being investigated.

Acknowledgements

The authors are grateful to Dr. Benyang Tang, Dr. Jingxi Lu, Dr. Adam Monahan and Yuval for helpful discussion, and to Dr. Ralf Giering for his TAMC compiler for adjoint codes. This work was supported by research and strategies grants to W. Hsieh from the Natural Sciences and Engineering Research Council of Canada.

Appendix A — Neural Network Model

A feed-forward neural network (NN) is a non-parametric statistical model for extracting nonlinear relations in the data. A common NN model configuration is to place between the input and output variables (also called 'neurons'), a layer of 'hidden neurons' (Fig.18). The value of the jth hidden neuron is

$$y_j = tanh(\sum_i w_{ij}x_i + b_j), \tag{17}$$

where x_i is the i-th input, w_{ij} the weight parameters and b_j the bias parameters. The hyperbolic tangent function is used as the transfer function (Bishop 1995, p127).

The output neuron is given by

$$z = \sum_j \tilde{w}_j y_j + \tilde{b} \tag{18}$$

A cost function

$$J = \langle (z - z^{obs})^2 \rangle \tag{19}$$

measures the mean square error between the model output z and the observed values z^{obs}. The parameters w_{ij}, $\tilde{w}_j$, b_j and $\tilde{b}$ are adjusted as the cost function is minimized by the optimization algorithm of Levenberg-Marquardt (Masters, 1995), without any constraints imposed. The procedure, known as network training, yields the optimal parameters for the network. The random number generator in MATLAB which generates uniformly distributed random numbers on the interval (0.0, 1.0) was used to initialize these parameters.

For an NN with m_1 inputs and m_2 hidden neurons, the number of model parameters is $m_1 \times m_2 + 2 \times m_2 + 1$. In this paper, we set the number of hidden neurons to 5, so the NN has 26 parameters.

The above is a brief description for a traditional feed-forward NN algorithm, used in Section 3. In later sections, the optimization will be constrained by the dynamical equations.

Appendix B — Discrete form of the hybrid model for coding

The computer code is directly generated from the discrete form of the hybrid model. The 4th-order Runge-Kutta is used to discretize the vector equation (7):

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{\mathbf{k1}_n}{6} + \frac{\mathbf{k2}_n}{3} + \frac{\mathbf{k3}_n}{3} + \frac{\mathbf{k4}_n}{6}, \tag{20}$$

$$\mathbf{k1}_n = h\mathbf{f}(\mathbf{v}_n, \mathbf{p}, t_n), \tag{21}$$

$$\mathbf{k2}_n = h\mathbf{f}(\mathbf{v}_n + 0.5\mathbf{k1}_n, \mathbf{p}, t_n + 0.5h), \tag{22}$$

$$\mathbf{k3}_n = h\mathbf{f}(\mathbf{v}_n + 0.5\mathbf{k2}_n, \mathbf{p}, t_n + 0.5h), \tag{23}$$

$$\mathbf{k4}_n = h\mathbf{f}(\mathbf{v}_n + \mathbf{k3}_n, \mathbf{p}, t_n + h), \tag{24}$$

where the subscript n indicates the time level, $\mathbf{f}$ $=(f1, f2, f3)^{tr}$, and f1, f2 and f3 are scalar functions:

$$f1 = -aX + aY, \tag{25}$$

$$f2 = -XZ + bX - Y, \tag{26}$$

$$f3 = \sum_j \tilde{w}_j [tanh(w_{1j}X + w_{2j}Y + w_{3j}Z + b_j)] + \tilde{b}, \tag{27}$$

j=1,2.. M (M=5, the number of hidden neurons).

Eq. (20) can be written as

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{K}_n \tag{28}$$

The Lagrange function of Eq. (10) is discretized as

$$L = J + \sum_{n=0}^{N-1} \mathbf{v}_n^{*tr}(\mathbf{v}_{n+1} - \mathbf{v}_n - \mathbf{K}_n), \tag{29}$$

$$J = \sum_{n=1}^{N} (\mathbf{v}_n - \mathbf{v}_n^{obs})^{tr} \mathbf{W}^{-1} (\mathbf{v}_n - \mathbf{v}_n^{obs}), \tag{30}$$

where $\mathbf{v}^*$ is the vector of Lagrange multipliers and its dimension is the same that of $\mathbf{v}$.

The equations for the best fit are obtained by requiring that all derivatives of L vanish:

$$\frac{\partial L}{\partial \mathbf{v}_n^*} = 0, \tag{31}$$

$$\frac{\partial L}{\partial \mathbf{v}_n} = 0, \tag{32}$$

$$\frac{\partial L}{\partial \mathbf{p}} = 0. \tag{33}$$

where the partial derivative with respect to a vector indicates a partial derivative with respect to each of the vector components. Eq.(31) gives nothing more than the (28); whereas (29) gives

$$\frac{\partial L}{\partial \mathbf{p}} = \sum_{n=0}^{N-1} (-\frac{\partial \mathbf{K}_n}{\partial \mathbf{p}}) \mathbf{v}_n^*. \tag{34}$$

The gradients from (34) are then provided to a quasi-Newton method to seek the optimal parameters.

The equations describing the trajectory of the Lagrange multipliers $\mathbf{v}_n^*$, i.e., the adjoint equations, can be obtained from ∂ L/∂ $\mathbf{v}_n$ and Eq. (29) (also see Anderson and Willebrand, 1989):

$$2 \sum_{n=1}^{N} \mathbf{W}^{-1} (\mathbf{v}_n - \mathbf{v}_n^{obs}) + \mathbf{v}_{n-1}^* - \mathbf{v}_n^* - (\frac{\partial \mathbf{K}_n}{\partial \mathbf{v}_n}) \mathbf{v}_n^* = 0, \qquad n = 1, ..., N, \tag{35}$$

$$\frac{\partial L}{\partial \mathbf{v}_0} = -\mathbf{v}_0^*, \tag{36}$$

$$\mathbf{v}_N^* = 0. \tag{37}$$

The $\partial \mathbf{K}_n / \partial \mathbf{v}_n$ and $\partial \mathbf{K}_n / \partial \mathbf{p}$ can be obtained by chain-rule, e.g.

$$\frac{\partial \mathbf{k}_n}{\partial \mathbf{v}_n} = \frac{1}{6} \frac{\partial \mathbf{k1}_n}{\partial \mathbf{v}_n} + \frac{1}{3} \frac{\partial \mathbf{k2}_n}{\partial \mathbf{k1}_n} \frac{\partial \mathbf{k1}_n}{\partial \mathbf{v}_n} + \frac{1}{3} \frac{\partial \mathbf{k3}_n}{\partial \mathbf{k2}_n} \frac{\partial \mathbf{k2}_n}{\partial \mathbf{k1}_n} \frac{\partial \mathbf{k1}_n}{\partial \mathbf{v}_n} + \frac{1}{6} \frac{\partial \mathbf{k4}_n}{\partial \mathbf{k3}_n} \frac{\partial \mathbf{k3}_n}{\partial \mathbf{k2}_n} \frac{\partial \mathbf{k2}_n}{\partial \mathbf{k1}_n} \frac{\partial \mathbf{k1}_n}{\partial \mathbf{v}_n}$$

Thus, the cycling procedure for computing the best fit of the hybrid model (4)-(6) to Lorenz model (1)-(3) is:

(i) Given initial guess values for the unknown parameters, using (28) to step the hybrid model forward in time from 1 to N to compute a first approximation to the Lorenz model.

(ii) Step equations (35) - (37) backward in time from N to 1 to compute approximate values of the Lagrange multipliers.

(iii) Evaluate the gradient of the cost function with respect to these parameters using Eq. (34) (and/or Eq. (36) if initial conditions need to be retrieved).

(iv) Use a descent algorithm to find the minimum of the cost function in the direction opposite to the gradient by a line search.

(v) Take results of the line search as the next guesses for the parameters and repeat the process starting from (i) until convergence criteria are satisfied.

The quasi-Newton method of Broyden-Fletcher-Goldfarb-Shanno ($BFGS$) (Press et al. 1992) is used in the above procedure for searching the optimal solutions. Fig. 19 is the variations of the normalized cost function and normalized gradient with the number of iterations for a weakly nonlinear case (Fig. 7) and a strongly nonlinear case (Fig. 11). The $BFGS$ algorithm has a memory requirement of $O(N^2)$ where N is the number of parameters involved in the optimization. In our case, the memory is 183 KB. For problems with larger N, the conjugate gradient method with a memory requirement of O(N) could be more suitable. A typical case here requires about 50 iteration for convergence, using about 3 minutes of CPU time on an SGI Origin 200 server.

In practice, the adjoint codes were not developed from Eqs. (35)-(37), but by the TAMC (Tangent and Adjoint Model Compiler, Giering and Kaminski, 1996) via the tangent linear equations. These equations are derived from Eq. (28):

$$\delta \mathbf{v}_{n+1} = \mathbf{A}_n \delta \mathbf{v}_n \tag{38}$$

where the matrix $\mathbf{A}_n$ is the tangent linear model at time-step n, and

$$\mathbf{A}_n = \mathbf{I} + \frac{\partial \mathbf{K}_n}{\partial \mathbf{v}_n} \tag{39}$$

with $\mathbf{I}$ the the identity matrix. The adjointness has been checked using the relation between the tangent-linear code $\mathbf{A}$ and its adjoint code $\mathbf{A}^*$ (Talagrand 1991; Navon et al. 1992):

$$\{\mathbf{a}, \mathbf{A}\mathbf{b}\} = \{\mathbf{A}^*\mathbf{a}, \mathbf{b}\} \tag{40}$$

where $\{,\}$ denotes the inner product, and **a** and **b** arbitrary vectors.

References

Anderson, D. L. T., and J. Willebrand (eds), 1989: *Oceanic Circulation Models: Combining Data and Dynamics*, Kluwer Academic Publishers, 287-302.

Barnett, T. P., M. Latif, N. E. Graham, M. Flugel, S. Pazan and W. White, 1993: ENSO and ENSO related predictability, Part1: Prediction of equatorial sea surface temperature with a hybrid coupled ocean-atmosphere model. *J.Climate*, **6,** 1545-1566.

Bennett, A. F., 1992: *Inverse Methods in Physical oceanography.* Cambridge University Press, 346pp.

Bishop, C. M., 1995: *Neural network for pattern recognition.* Clarendon Press . Oxford, 482pp.

Courtier P., E. Anderson, W. Heckley, J. Pailleux, D. Vasiljevic, M. Hamrud, A. Hollingsworth, E. Rabier, and M. Fisher, 1998: The ECMWF implementation of three-dimensional variational assimilation (3D-VAR)-I-Formulation. *Quart. J. Roy. Met. Soc.*, **124,** 1783-1807.

Cybenko, G., 1989: Approximation by superposition of a sigmoidal function. *Math. Control, Signal, Syst.*, **2,** 303-314.

Daley, R., 1991: *Atmospheric data analysis*, Cambridge atmospheric and Space Science series, Cambridge University Press, 457pp.

Elsner, J. B., and A. A. Tsonis, 1992: Nonlinear prediction, chaos and noise. *Bull. Amer. Meteor. Soc.*, **73,** 49-60.

Evensen, G., 1997: Advanced Data Assimilation for strongly nonlinear dynamics. Mon. Wea. Rev., **125,** 1342-1354

Gauthier, P., 1992: Chaos and quadric-dimensional data assimilation: a study based on the Lorenz model. *Tellus*, **44A,** 2-17.

Gauthier, P., C. Charette, L. Fillion, P. Koclas and S. Laroche, 1999: Implementation of a 3D variational data assimilation system at the Canadian Meteorological Centre. Part I: The global analysis. *Atmosphere-Ocean*, **37,** 103-156.

Giering, R., and Kaminski,T., 1998: Recipies for Adjoint Code Construction, *ACM TOMS*, **24**(4), 437-474.

Ghil, M. and P. Malanotte-Rizzoli, 1991: Data Assimilation in Meteorology and Oceanography. *Advances in Geophysics*, **33,** 141-265.

Hannachi, A., and B. Legras, 1995: Simulated annealing and weather regimes classification. *Tellus*, **47A,** 955-973.

Hsieh, W. W., and B. Tang, 1998: Applying neural network models to prediction and analysis in meteorology and oceanography. *Bull. Amer. Meteor. Soc.*, **79,** 1855-1870.

Ji, M., D. W. Behringer, and A. Leetmaa, 1998: An improved coupled model for ENSO prediction and implications for ocean initialization-PART II-The coupled model. *Mon. Wea. Rev.*, **126,** 1022-1034.

Kruger, J., 1993: Simulated Annealing - A Tool for Data Assimilation into an Almost Steady Model State. *J. Phys. Oceanogr.*, **23,** 679-688

Lacarra, J. F., and O. Talagrand, 1988: Short-range evolution of small perturbations in a barotropic model. *Tellus*, **40A,** 81-95.

Le Dimet, F., and O. Talagrand, 1986: Variational algorithms for analysis and assimilation of meteorology observations: theoretical aspects. *Tellus*, **38A,** 97-110.

Lorenz, E. N., 1963: Deterministic nonperiodic flow. *J. Atmos. Sci.*, **20,** 130-141.

Lu, J., and W. W. Hsieh, 1998: On determining initial conditions and parameters in a simple coupled model by adjoint data assimilation. *Tellus*, **50A,** 534-544.

Masters, T., 1995: *Advanced Algorithms for neural network—A C++ source book.* John Wiley & Sons, Inc., 431pp.

Miller, R. N., M. Ghil and F. Gauthiez, 1994: Advanced data assimilation in strongly nonlinear dynamical systems. *J. Atmos. Sci.*, **51,** 1037-1056.

—, and —, 1990: Data assimilation in strongly nonlinear current system. *Proc. Int. Assimilation of Observation in Meteorology and Oceanography*, pp93-98, 9-13 July 1990, Clermont-Ferrand, France.

Monahan, A. H., 2000: Nonlinear principal component analysis by neural networks: theory and application to the Lorenz system. *J. Climate*, **13,** 821-835.

Navon, I. M., X. Zhou, J. Derber and J. Sela, 1992: Variational data assimilation with an adiabatic version of the NMC spectral model. *Mon. Wea. Rev.*, **120,** 1433-1446.

—, 1998: Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography. *Dynamics of Atmospheres and Oceans*, **27** (1-4), 55-79.

Palmer, T. N., 1993: Extended-Range atmospheric prediction and the Lorenz model. *Bull. Amer. Meteor. Soc.*, **74**(1), 49-65.

Press, W. H., S.A. Teukolsky, W. T. Vetterling, and B. P. Flannery: 1992: *Numerical recipes in Fortran, Second edition*, Cambridge University Press, 963pp.

Syu, H. H., J. D. Neelin, and D. Gutzler, 1995: Seasonal and Interannual variability in a hybrid coupled GCM, *J. Climate*, **9,** 2121-2143

Talagrand, O., 1991: The use of adjoint equations in numerical modelling of the atmospheric circulation. In Andreas Griewank and George F. Corliess, editors, *Automatic Differentiation of Algorithms: Theory, Implementation and Application.* 169-180, SIAM.

Tang, Y., and W. W. Hsieh, B. Tang and K. Haines, 1999: A Neural Network Atmospheric Model for Hybrid Coupled Modelling, *Climate Dynamics* (in press)

Tziperman, E., and W. C. Thacker, 1989: An optimal control/adjoint equations approach to studying the oceanic general circulation. *J. Phys. Oceanogr.*, **19,** 1471-1485.

Wilks, D. S., 1995: *Statistical Methods in the Atmospheric Sciences. San Diego*, Academic Press, 467 pp.

Zhu, Y., and I. M. Navon, 1999: Impact of Key Parameters Estimation on the performance of the FSU spectral model using the full physics adjoint. *Mon. Wea. Rev.*, **127,** 1497-1517.

Figure captions

Fig. 1. The Lorenz system integrated over 15000 time steps for the weakly nonlinear case (left column) and the highly nonlinear case (right column).

Fig. 2. The ensemble mean of 25 NNs (dot-dash curve) for approximating the third Lorenz equation (solid curve) for the weakly nonlinear case (left column) and the highly nonlinear case (right column). Top panels (a) and (b) are for the training period of 3000 time steps, and bottom panels (c) and (d) for test period of 1000 time steps.

Fig. 3. The hybrid model (dot-dash curve) and the true Lorenz model (solid curve) integrated from identical initial conditions for the weakly nonlinear case (left column) and the highly nonlinear case (right column).

Fig. 4. Schematic diagram for the (a) SC (Strong Continuity constraint), (b) NC (No Continuity constraint) and (c) PSC (Partial Strong Continuity constraint) assimilation schemes. The solid curves are the model trajectory, dots, the observed values and dash lines, the model errors to be minimized. Here a model trajectory starts from an observed value.

Fig. 5. The NC assimilation results, with the model training results shown on the left panels, and predictions over the test period on the right panels; where the solid curves show the true data and dot-dash curves the outputs of the hybrid model. The dot-dash curves overlap the solid curves in the left panels.

Fig. 6. The PSC assimilation results with $AS=$ 200 time steps. The model training results are shown on the left, and model predictions on the right, with solid curves for the true data and dot-dash curves for the outputs of the hybrid model.

Fig. 7. Same as Fig. 6 but for $AS=$ 1000 time steps.

Fig. 8. Correlation skills (averaged over 10 experiments) for the variables X, Y and Z at various assimilation windows (T = 100, 300, 500 and 800 time steps) during (a) the training period, and (b) the test period.

Fig. 9. (a) Correlation skills and (b) REE for the variables X, Y and Z during the training and test periods (averaged over 100 experiments), for the weakly nonlinear case, where the SC assimilation retrieves the NN parameters. The error bars show the standard deviation. Results from the simple hybrid model without data assimilation (Section 3) are also shown for comparison.

Fig. 10. Bar charts from 100 assimilation experiments for the highly nonlinear case showing the number of experiments with (a) correlation above a particular correlation level and (b) REE below a particular REE level, during the test period.

Fig. 11. A typical experiment with class 1 results for the highly nonlinear case, showing (a) the fitting during the training period (left panels) and (b) the forecasting during the testing period (right panels). Solid curves denote the true value, while dot- dashed curves denote model results.

Fig. 12. Same as Fig.11 but for a typical experiment with class 2 (i.e. moderately successful forecast) results.

Fig. 13. Same as Fig.11, but for a typical experiment with class 3 (i.e. unsuccessful forecast) results.

Fig. 14. The system trajectories in the $X - Y - Z$ space during the training period (thin curves) and the testing period (thick curves) for (a) the class 1 experiment and (b) the class 3 experiment.

Fig. 15. The results from retrieving the dynamical parameters a, b and the initial conditions of the hybrid model, where the initial guesses were the true values scaled down by 20%.

Fig. 16. The correlation skill averaged over 100 experiments where the NN parameters, the dynamical parameters a and b, and the initial conditions were retrieved for the weakly nonlinear case. Error bars indicate the standard deviation.

Fig. 17. Bar charts from 100 assimilation experiments for the highly nonlinear case showing the number of experiments with (a) correlation above a particular correlation level and (b) REE below a particular REE level, during the test period. The NN parameters, the dynamical parameters a and b, and the initial conditions were retrieved.

Fig. 18. An example of a neural network model, where there are three neurons in the input layer, five in the hidden layer, and one in the output layer. The parameters w_{ij} and $\tilde{w}_j$ are the weights, and b_j and $\tilde{b}$ are the biases. The parameters b_j and $\tilde{b}$ can also be regarded as the weights for constant inputs of value 1.

Fig. 19. Variations of the normalized cost function (J/J_0) (solid line) and normalized gradient $(\|\mathbf{g}\|\|\mathbf{g_0}\|^{-1})$ (dash line) with the number of iterations for (a) the weakly nonlinear case of Fig. 7 and (b) the strongly nonlinear case of Fig. 11. The better initial guesses in (a) meant that the cost function and its gradients had relatively less distance to drop to convergence than in (b).

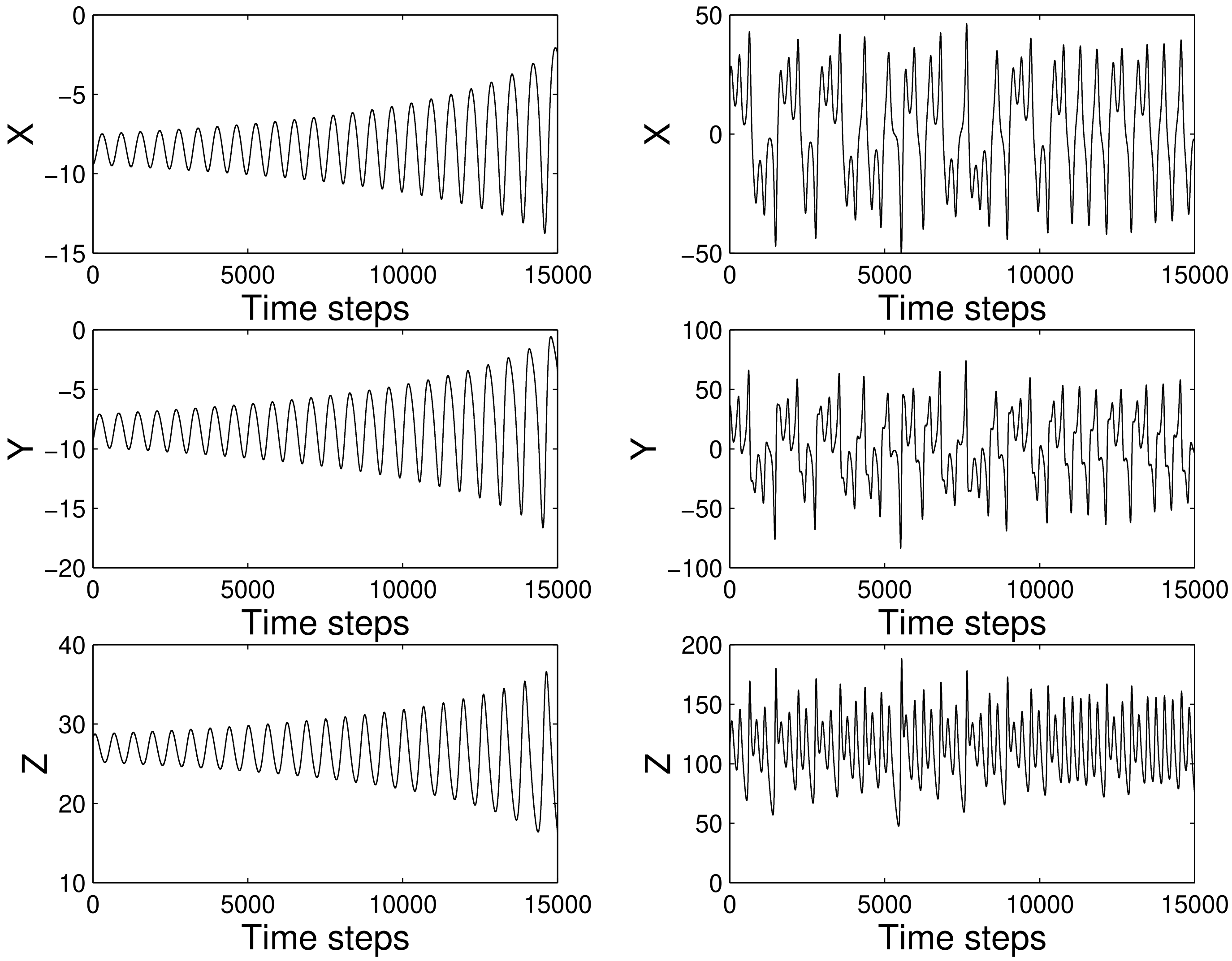


Figure 1: The Lorenz system integrated over 15000 time steps for the weakly nonlinear case (left column) and the highly nonlinear case (right column).

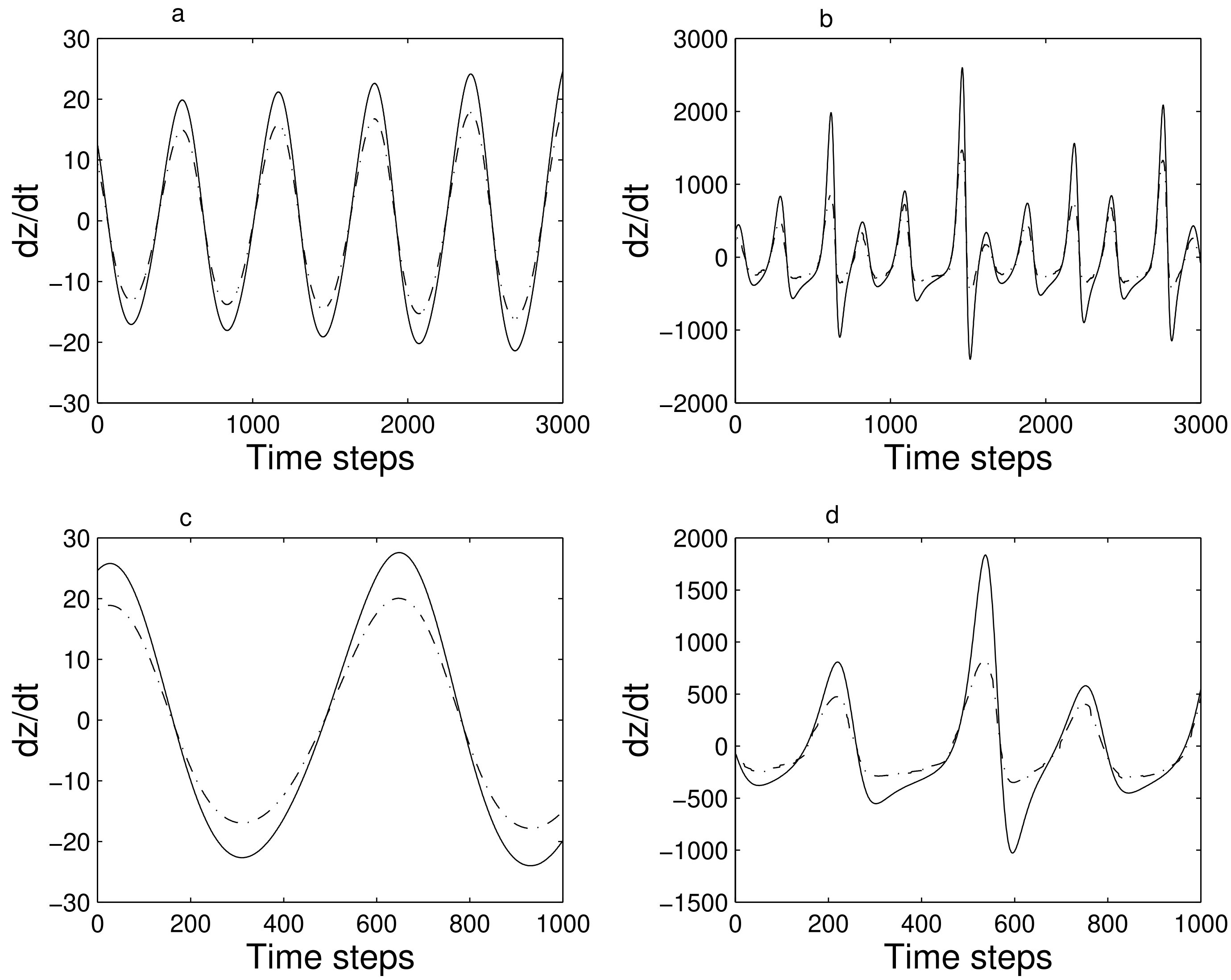


Figure 2: The ensemble mean of 25 NNs (dot-dash curve) for approximating the third Lorenz equation (solid curve) for the weakly nonlinear case (left column) and the highly nonlinear case (right column). Top panels (a) and (b) are for the training period of 3000 time steps, and bottom panels (c) and (d) for test period of 1000 time steps.

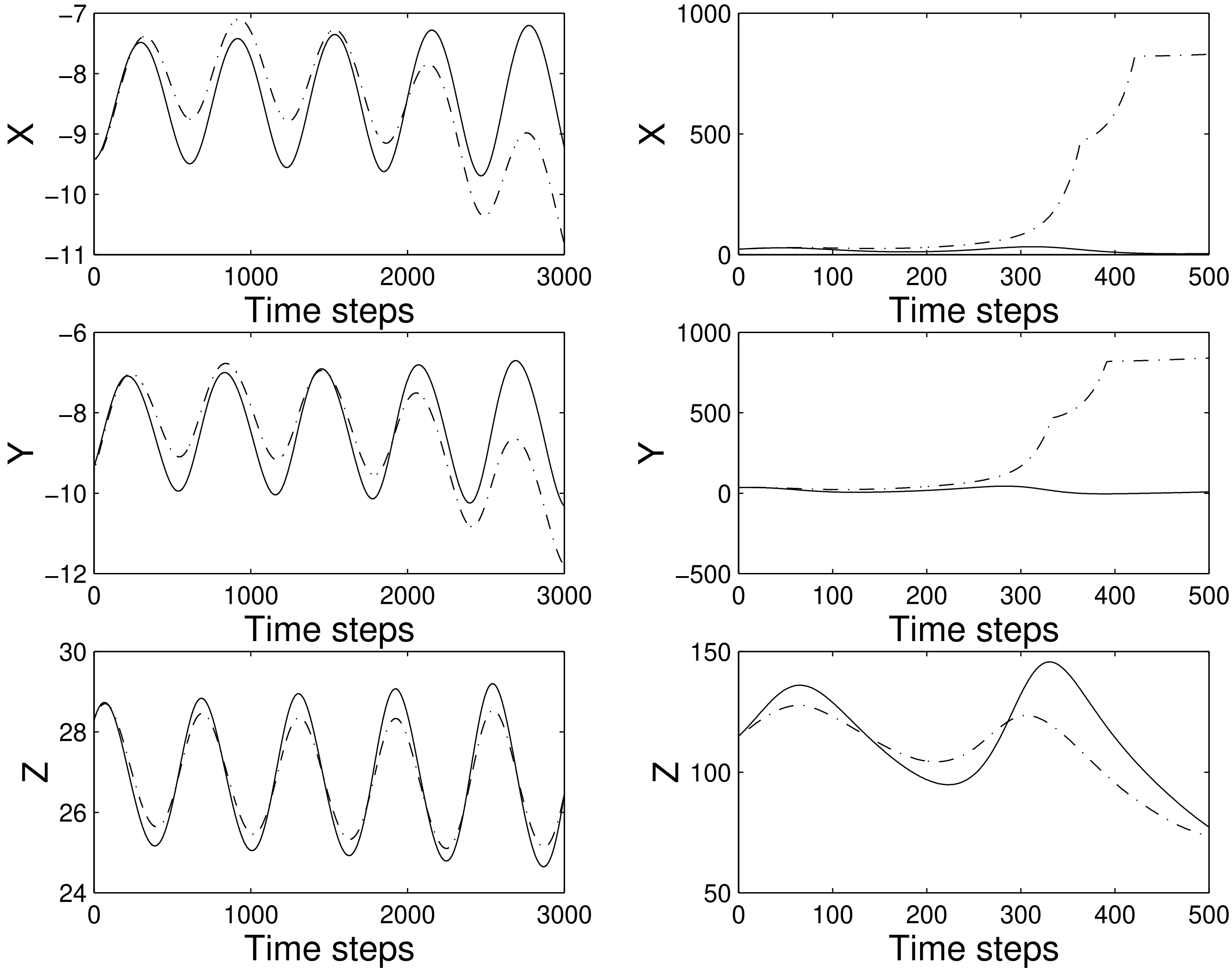


Figure 3: The hybrid model (dot-dash curve) and the true Lorenz model (solid curve) integrated from identical initial conditions for the weakly nonlinear case (left column) and the highly nonlinear case (right column).

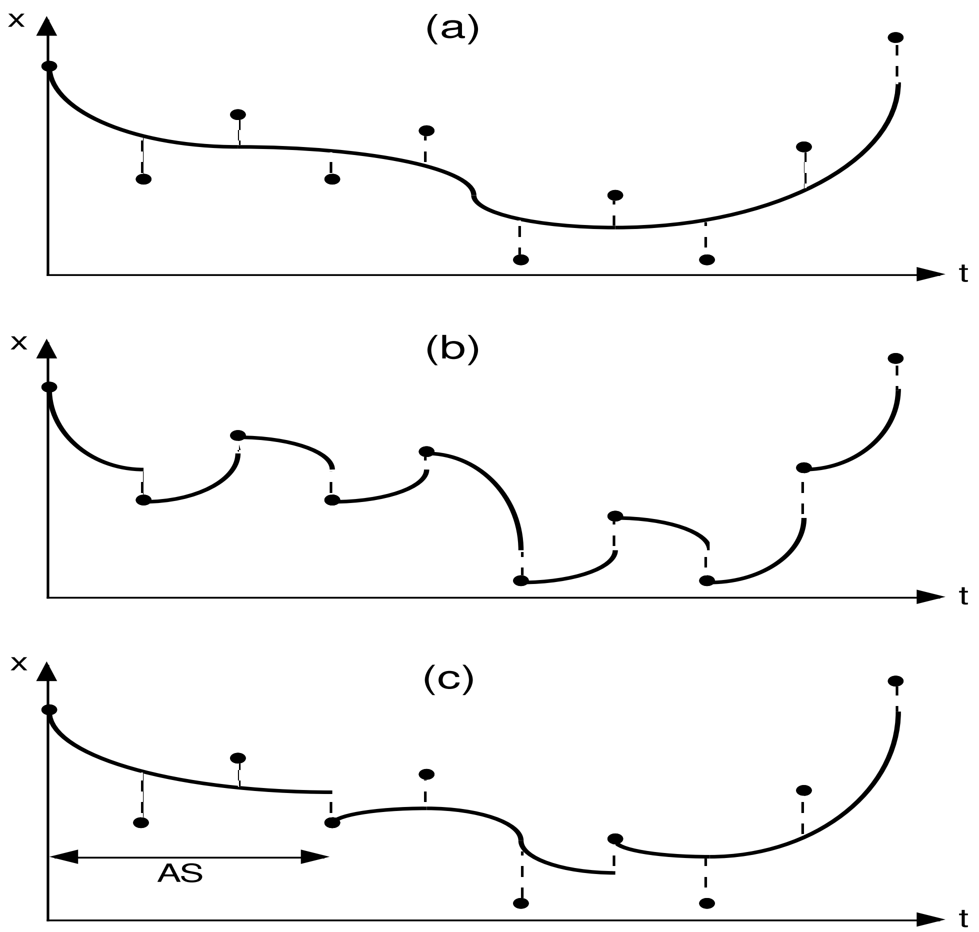


Figure 4: Schematic diagram for the (a) SC (Strong Continuity constraint), (b) NC (No Continuity constraint) and (c) PSC (Partial Strong Continuity constraint) assimilation schemes. The solid curves are the model trajectory, dots, the observed values and dash lines, the model errors to be minimized. Here a model trajectory starts from an observed value.

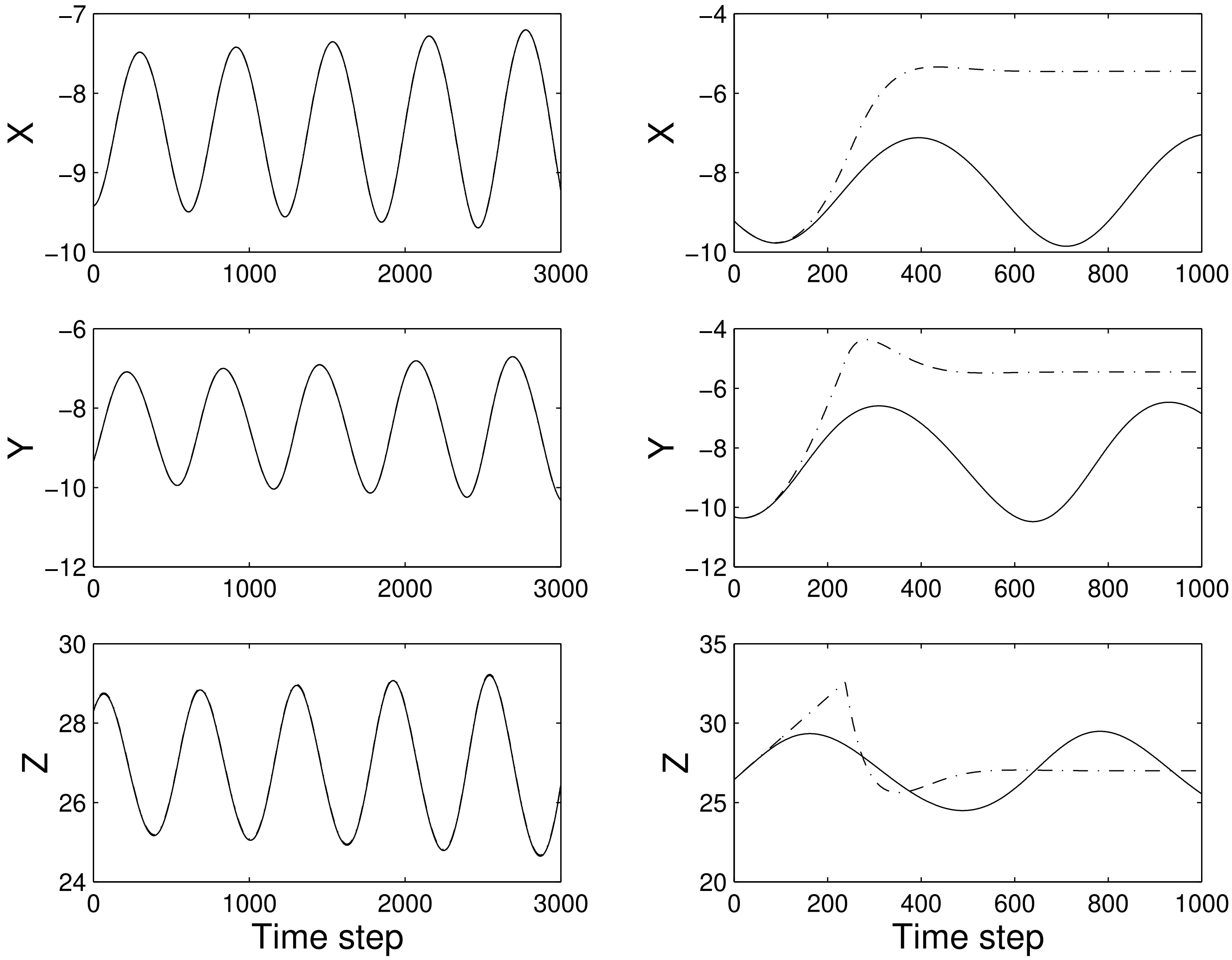


Figure 5: The NC assimilation results, with the model training results shown on the left panels, and predictions over the test period on the right panels; where the solid curves show the true data and dot-dash curves the outputs of the hybrid model. The dot-dash curves overlap the solid curves in the left panels.

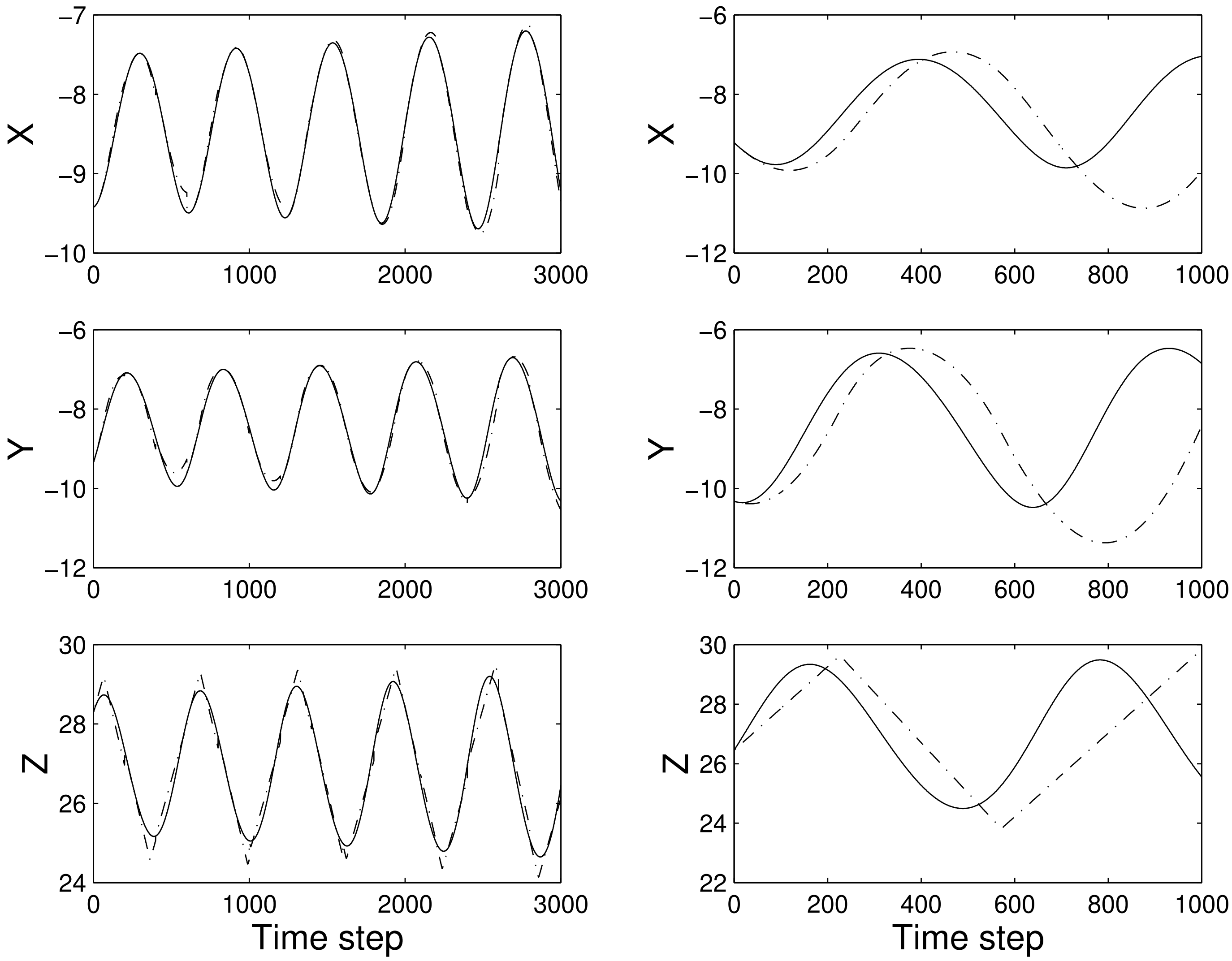


Figure 6: The PSC assimilation results with $AS=$ 200 time steps. The model training results are shown on the left, and model predictions on the right, with solid curves for the true data and dot-dash curves for the outputs of the hybrid model.

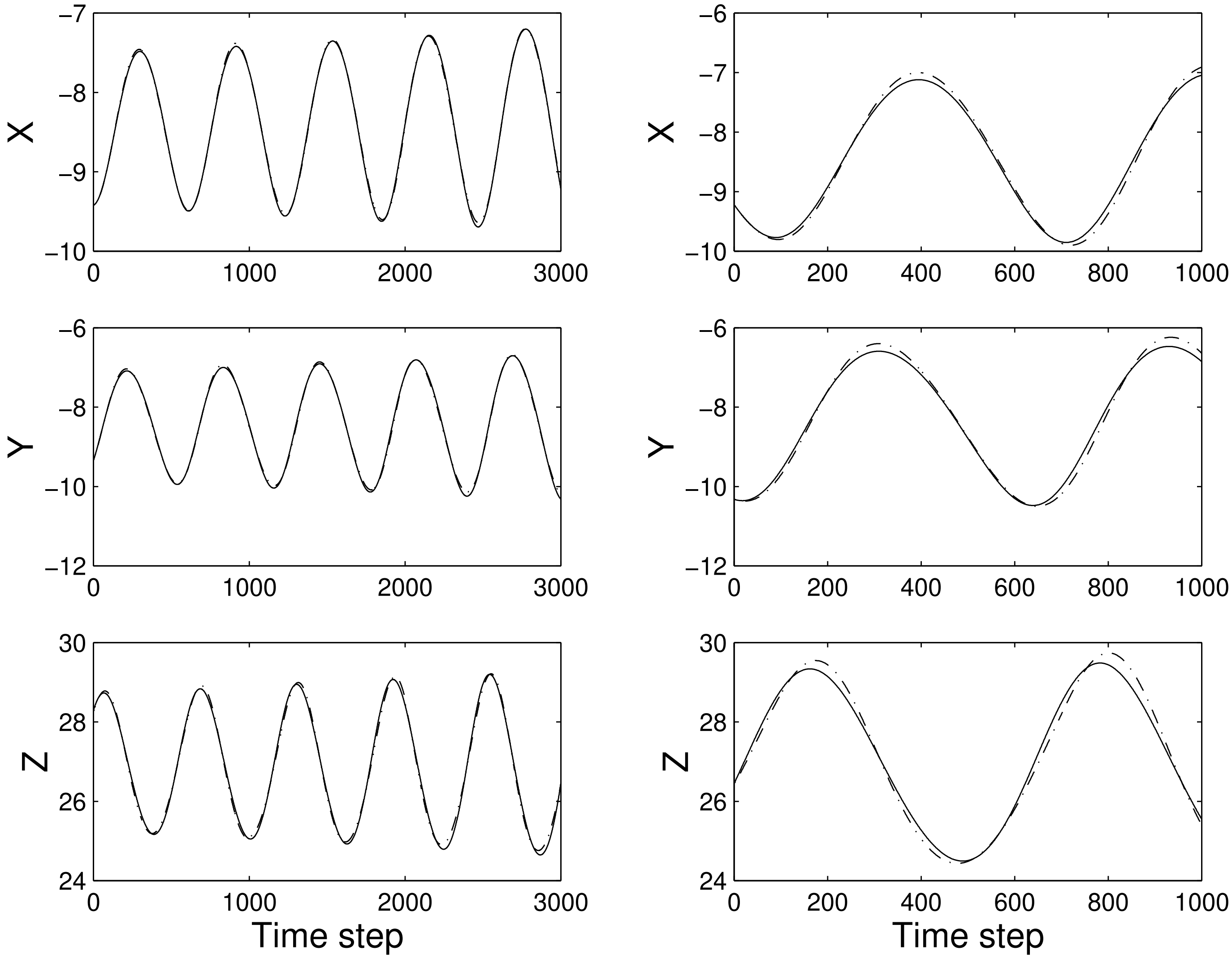
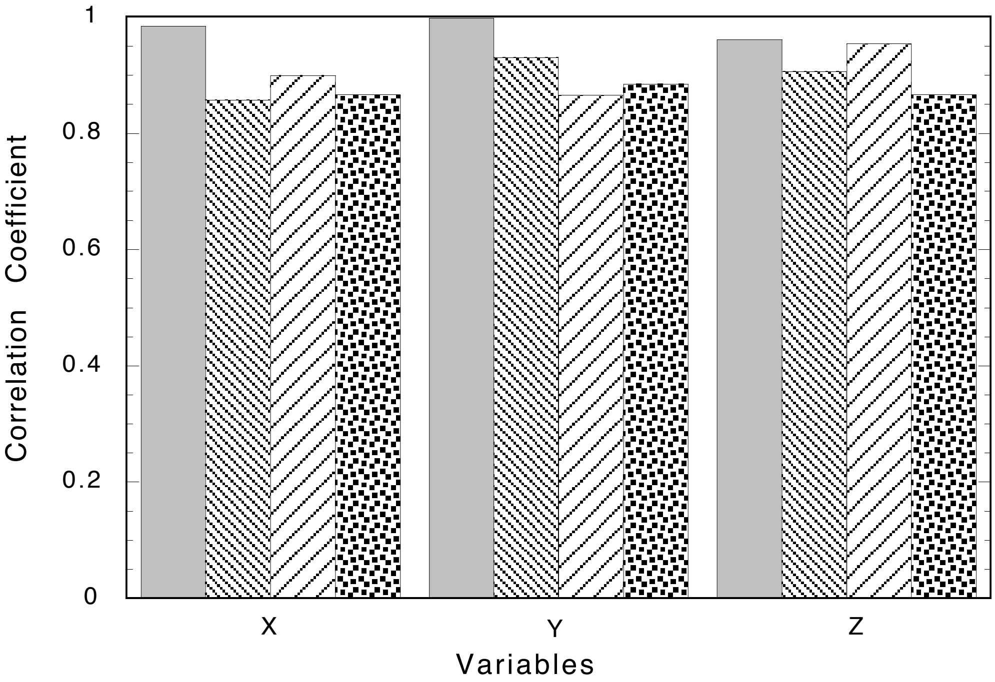


Figure 7: Same as Fig. 6 but for AS= 1000 time steps.

a



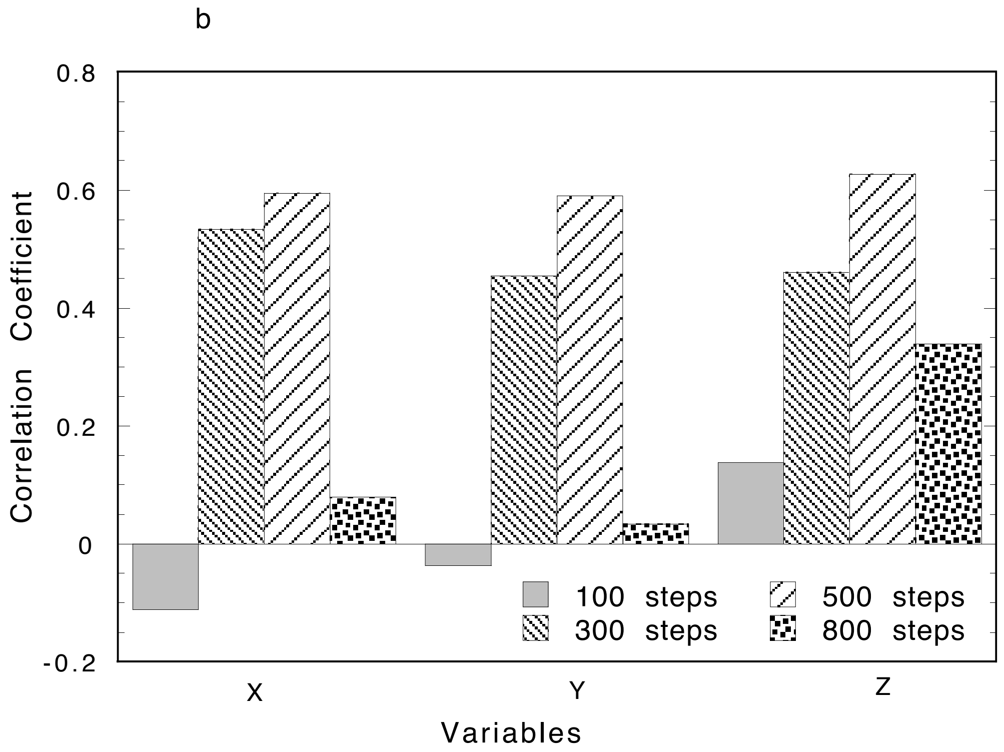
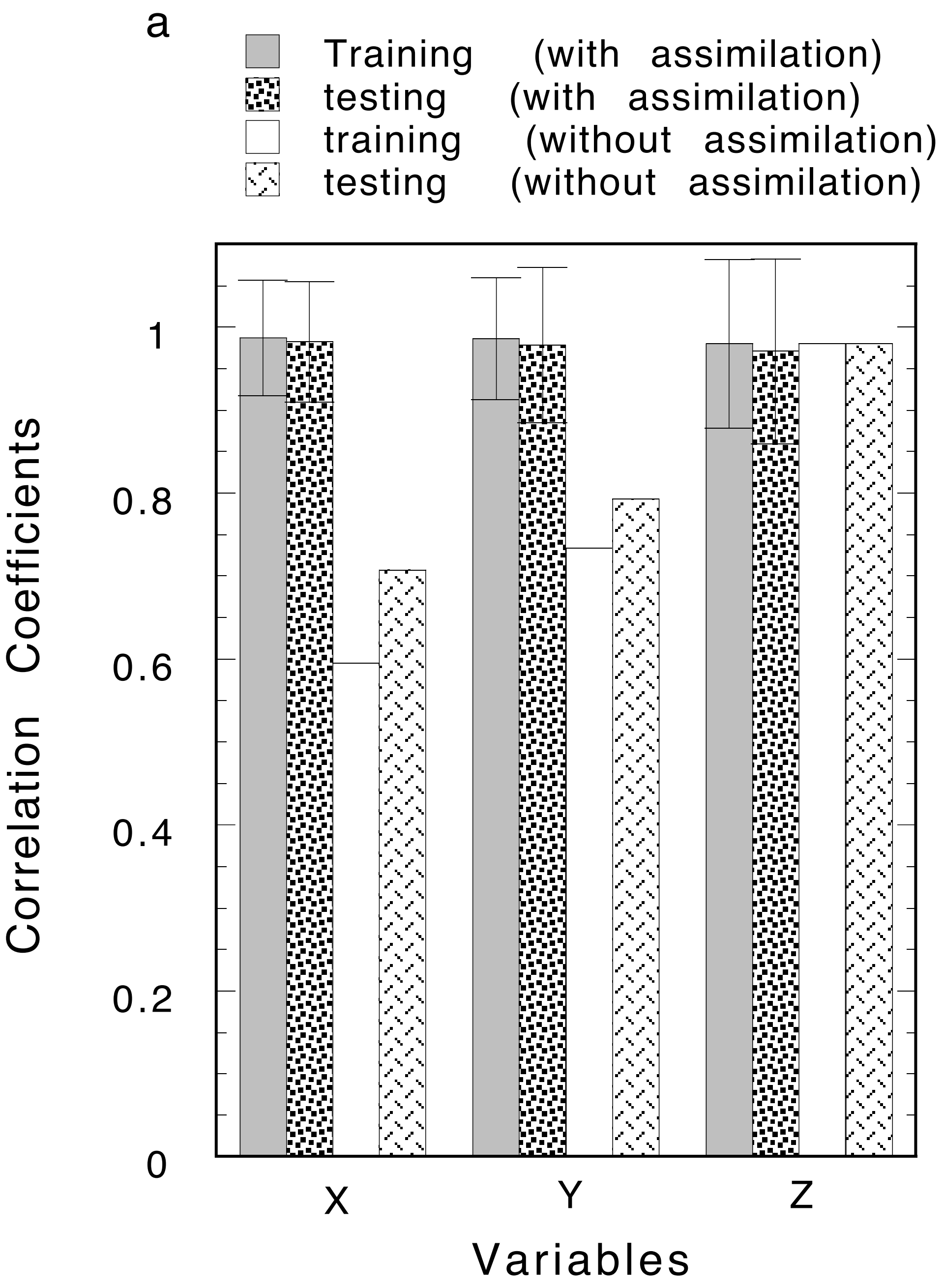


Figure 8: Correlation skills (averaged over 10 experiments) for the variables X, Y and Z at various assimilation windows (T = 100, 300, 500 and 800 time steps) during (a) the training period, and (b) the test period.


a
Training (with assimilation)
testing (with assimilation)
training (without assimilation)
testing (without assimilation)
Correlation Coefficients
0
0.2
0.4
0.6
0.8
1
X
Y
Z
Variables

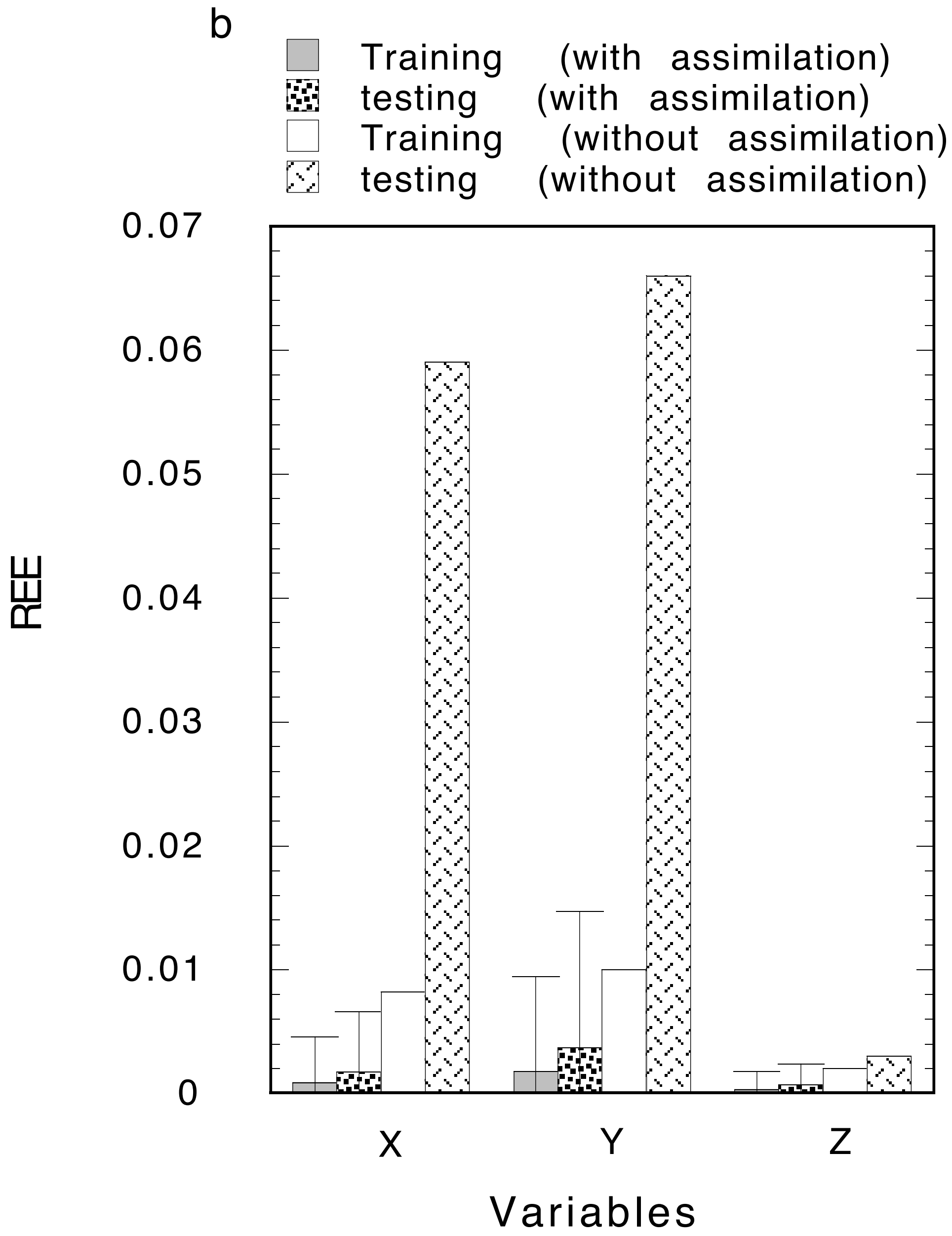
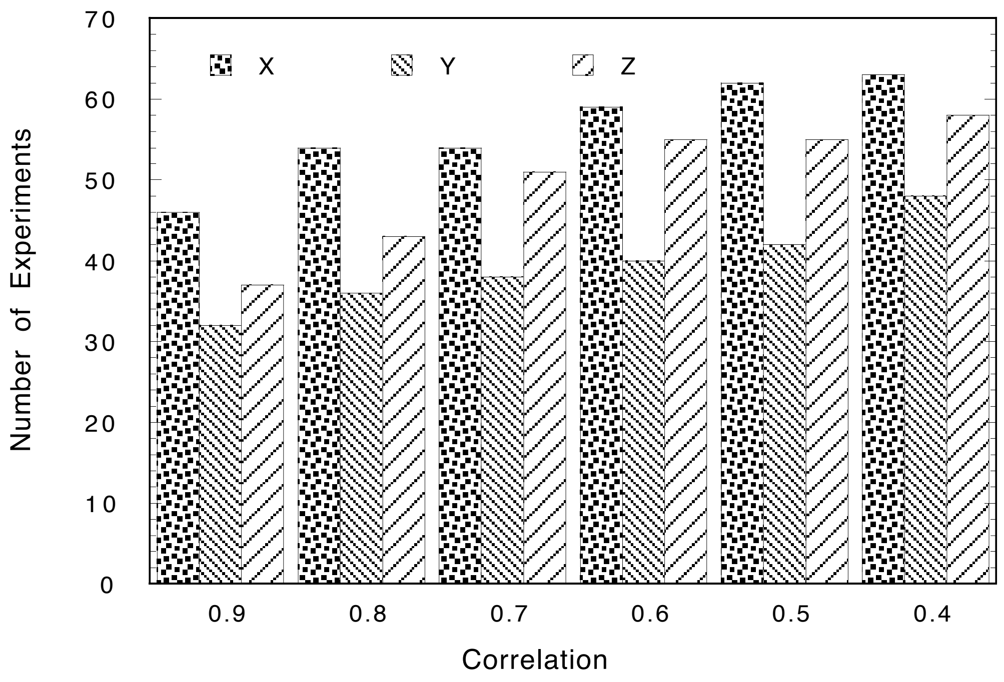


Figure 9: (a) Correlation skills and (b) REE for the variables X, Y and Z during the training and test periods (averaged over 100 experiments), for the weakly nonlinear case, where the SC assimilation retrieves the NN parameters. The error bars show the standard deviation. Results from the simple hybrid model without data assimilation (Section 3) are also shown for comparison.

a


X
Y
Z
Number of Experiments
70
60
50
40
30
20
10
0
0.9
0.8
0.7
0.6
0.5
0.4
Correlation

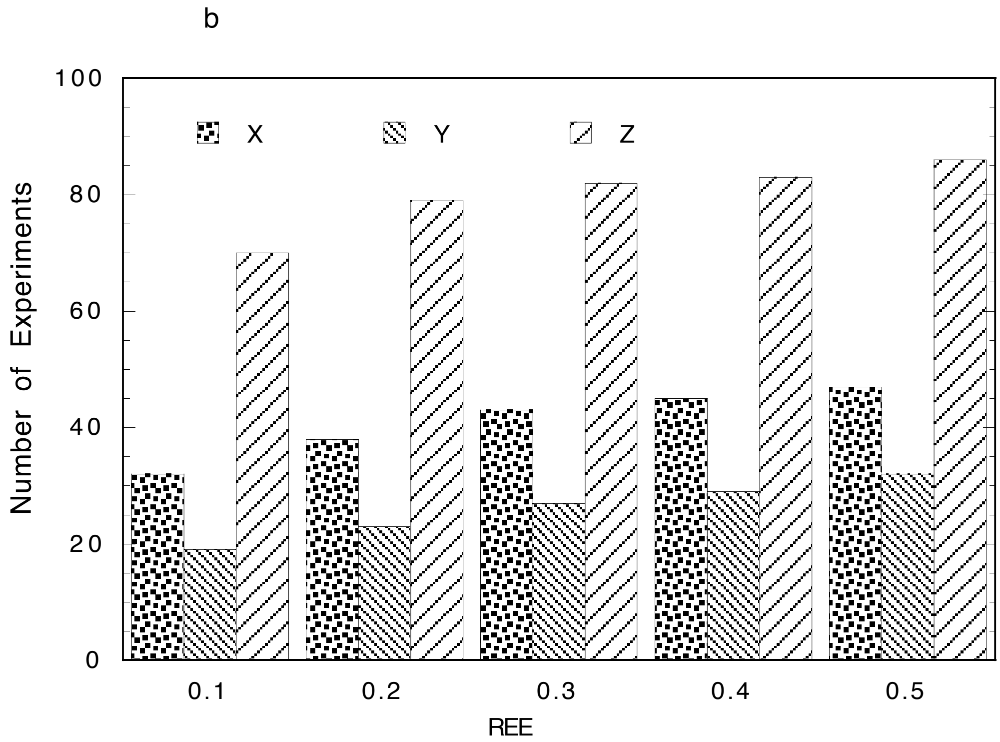


Figure 10: Bar charts from 100 assimilation experiments for the highly nonlinear case showing the number of experiments with (a) correlation above a particular correlation level and (b) REE below a particular REE level, during the test period.

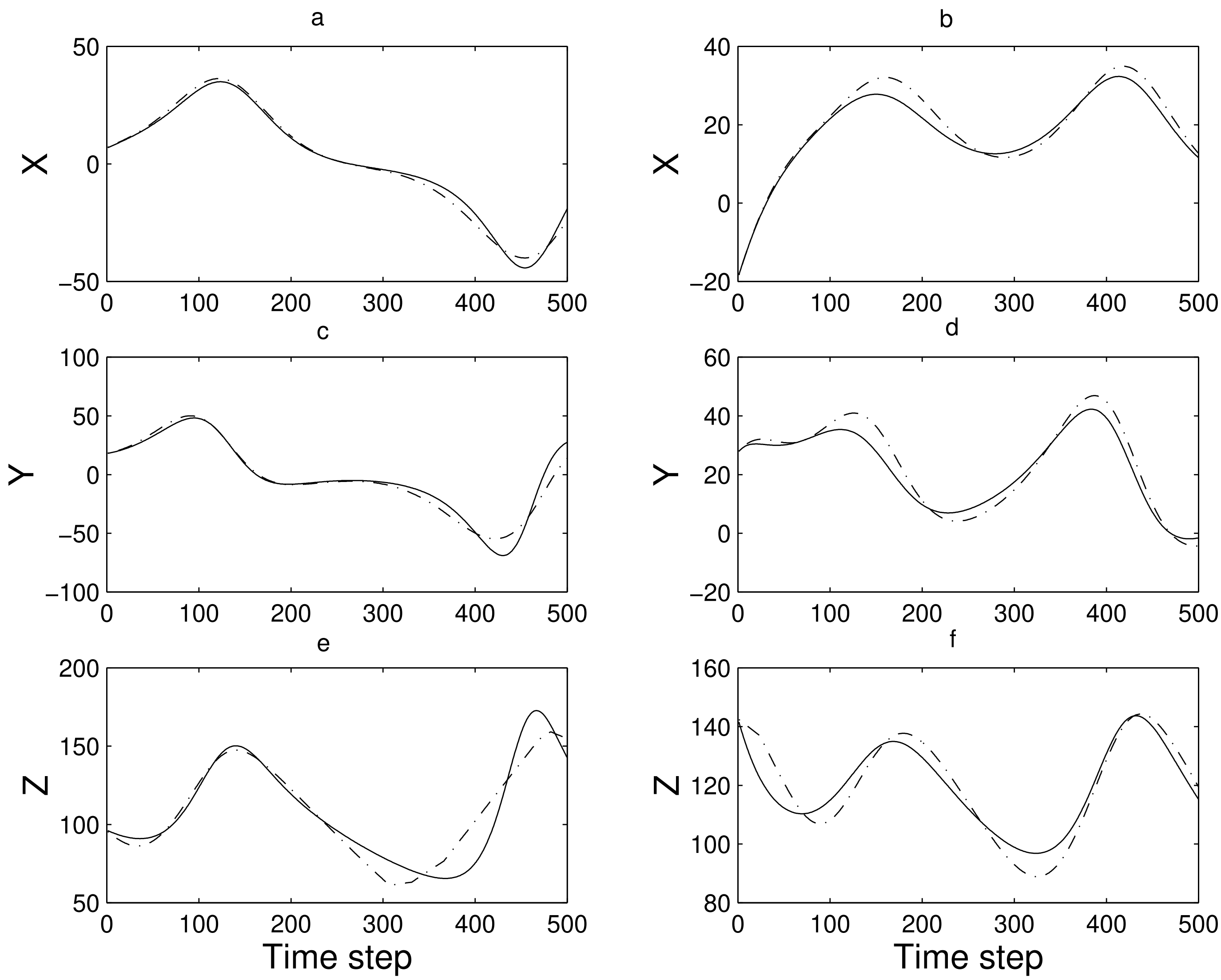


Figure 11: A typical experiment with class 1 results for the highly nonlinear case, showing (a) the fitting during the training period (left panels) and (b) the forecasting during the testing period (right panels). Solid curves denote the true value, while dot- dashed curves denote model results.

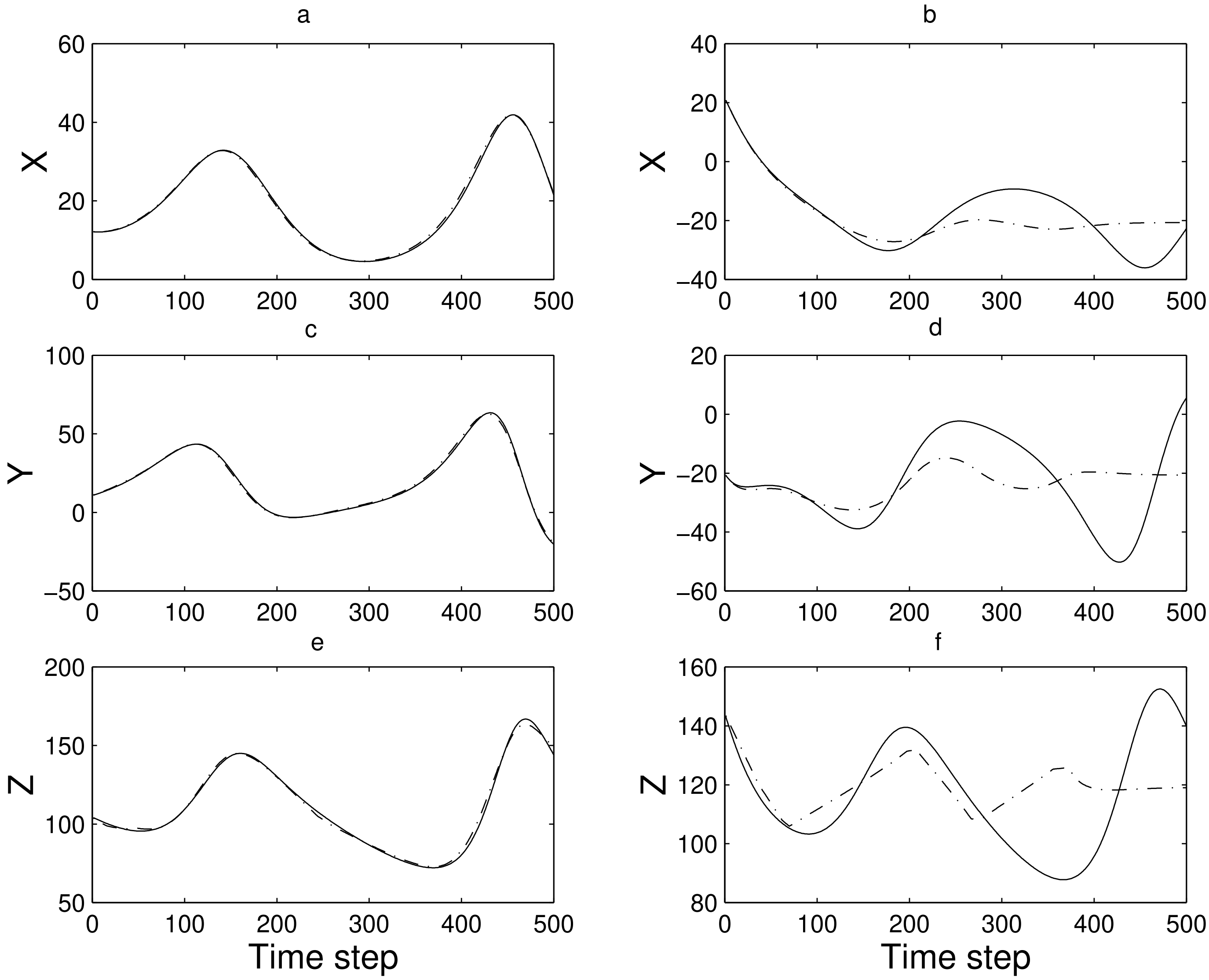


Figure 12: Same as Fig.11 but for a typical experiment with class 2 (i.e. moderately successful forecast) results.

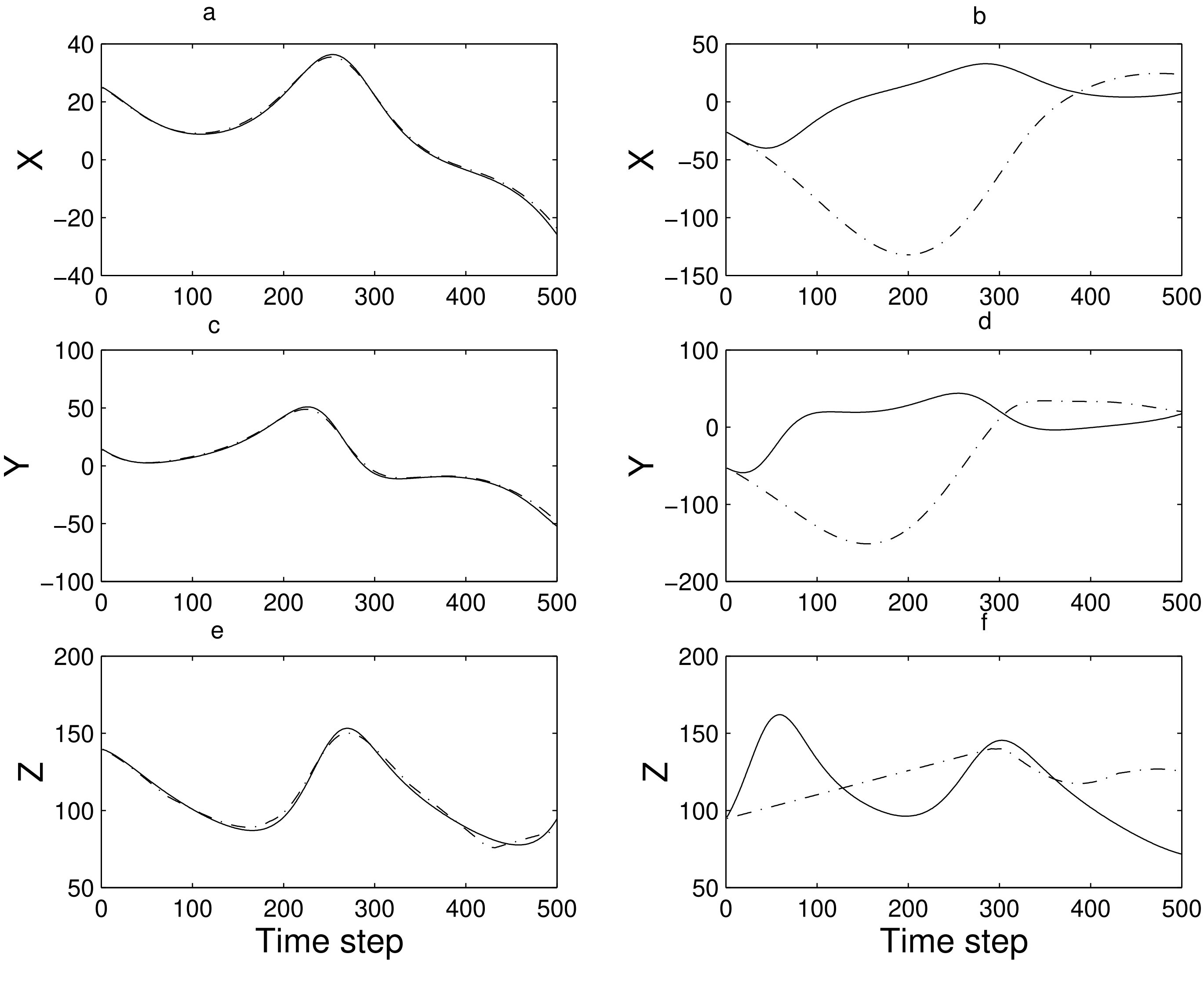


Figure 13: Same as Fig.11, but for a typical experiment with class 3 (i.e. unsuccessful forecast) results.

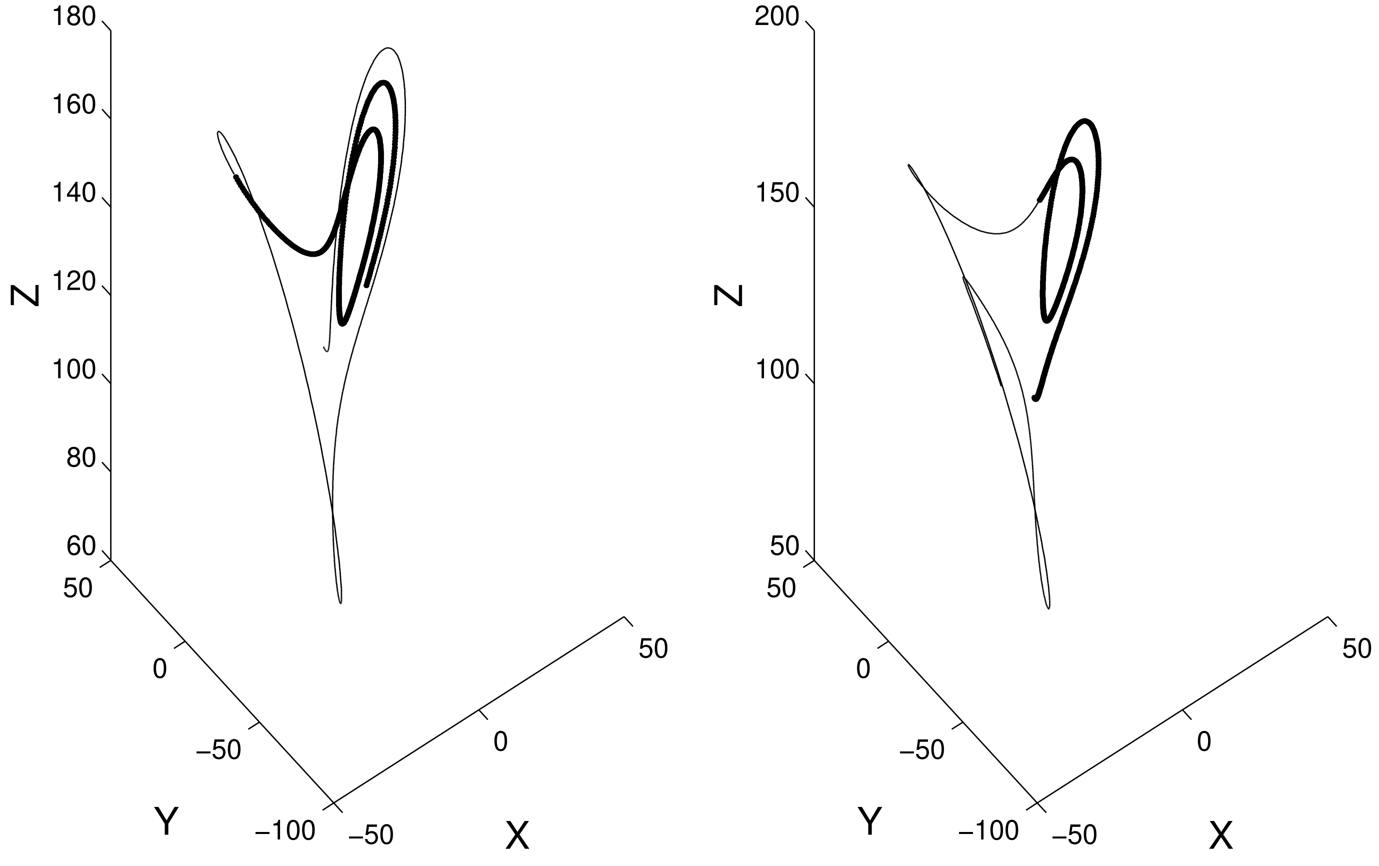


Figure 14: The system trajectories in the $X - Y - Z$ space during the training period (thin curves) and the testing period (thick curves) for (a) the class 1 experiment and (b) the class 3 experiment.

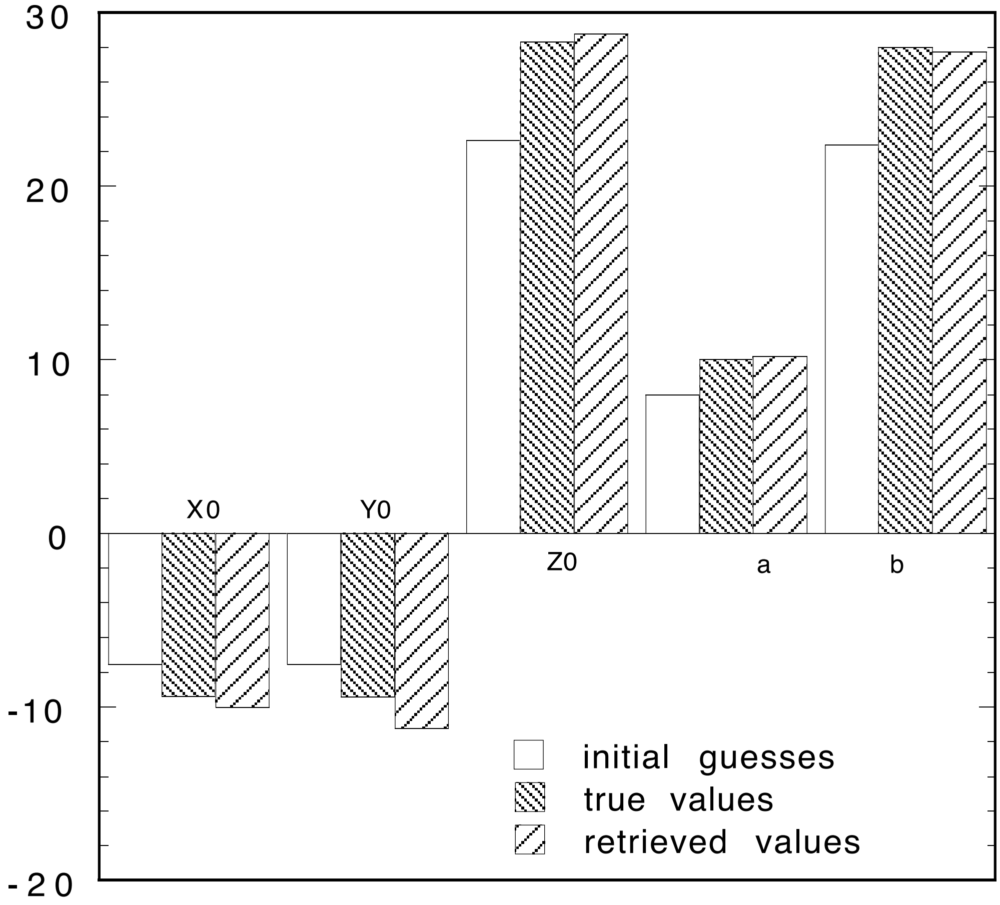


Figure 15: The results from retrieving the dynamical parameters a, b and the initial conditions of the hybrid model, where the initial guesses were the true values scaled down by 20%.

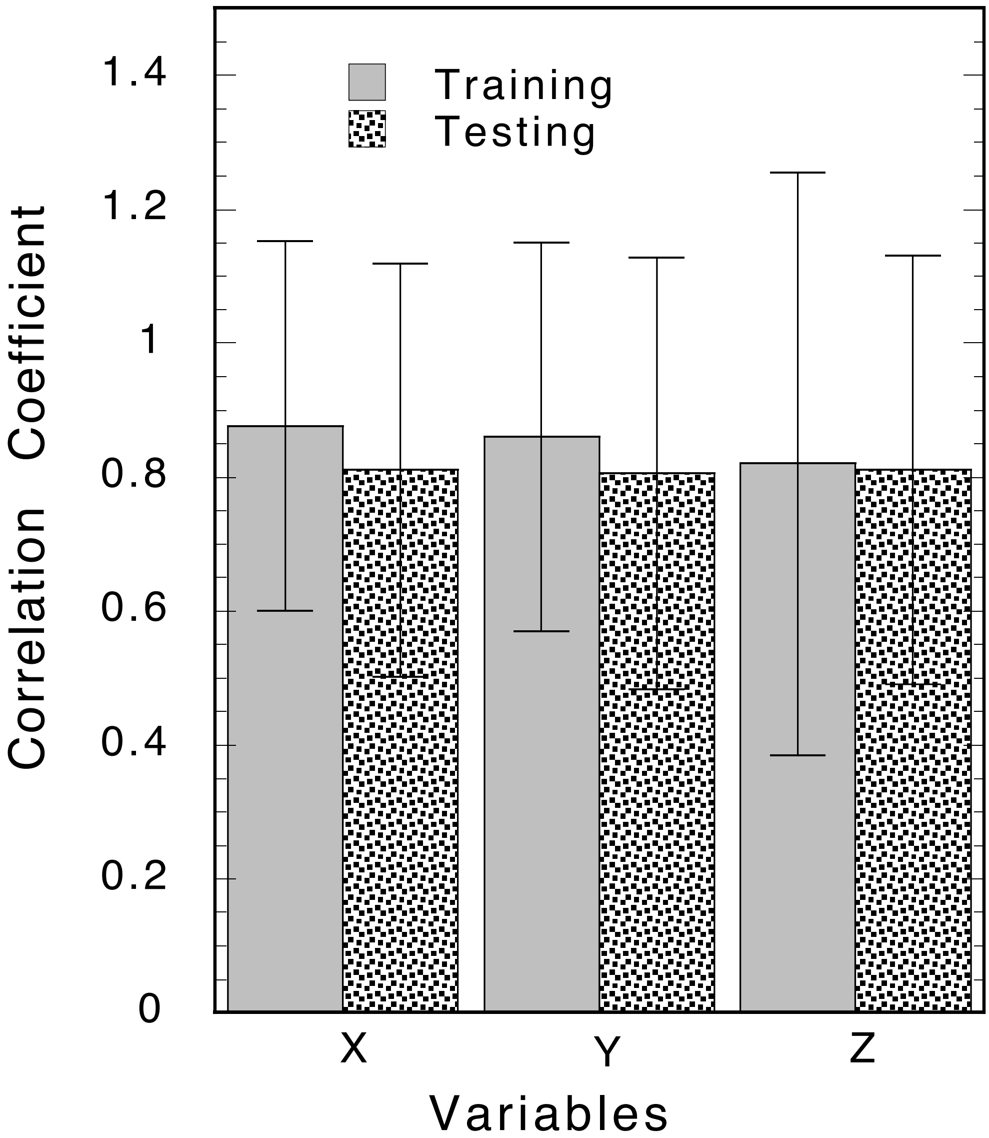
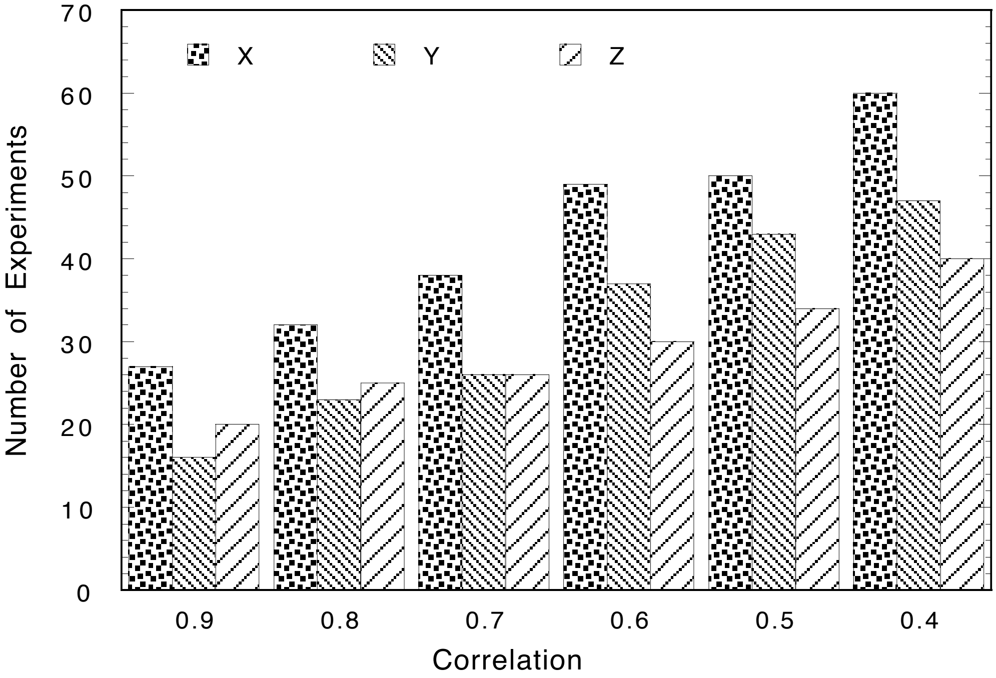


Figure 16: The correlation skill averaged over 100 experiments where the NN parameters, the dynamical parameters a and b, and the initial conditions were retrieved for the weakly nonlinear case. Error bars indicate the standard deviation.

a



70
60
50
40
30
20
10
0
Number of Experiments
X
Y
Z
0.9
0.8
0.7
0.6
0.5
0.4
Correlation

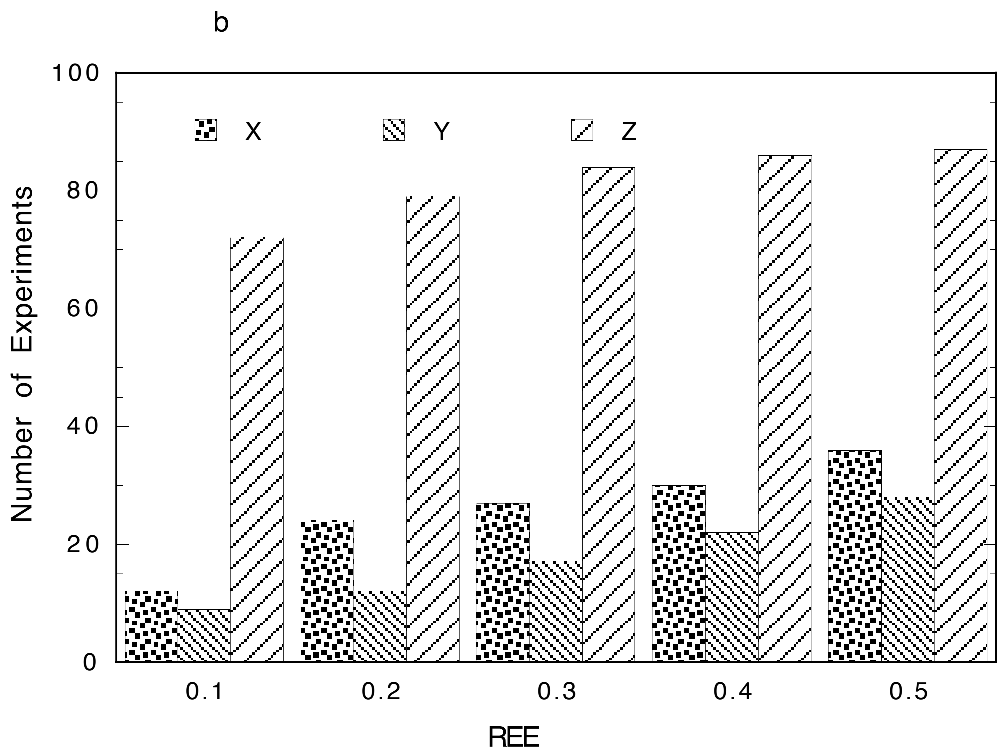


Figure 17: Bar charts from 100 assimilation experiments for the highly nonlinear case showing the number of experiments with (a) correlation above a particular correlation level and (b) REE below a particular REE level, during the test period. The NN parameters, the dynamical parameters a and b, and the initial conditions were retrieved.

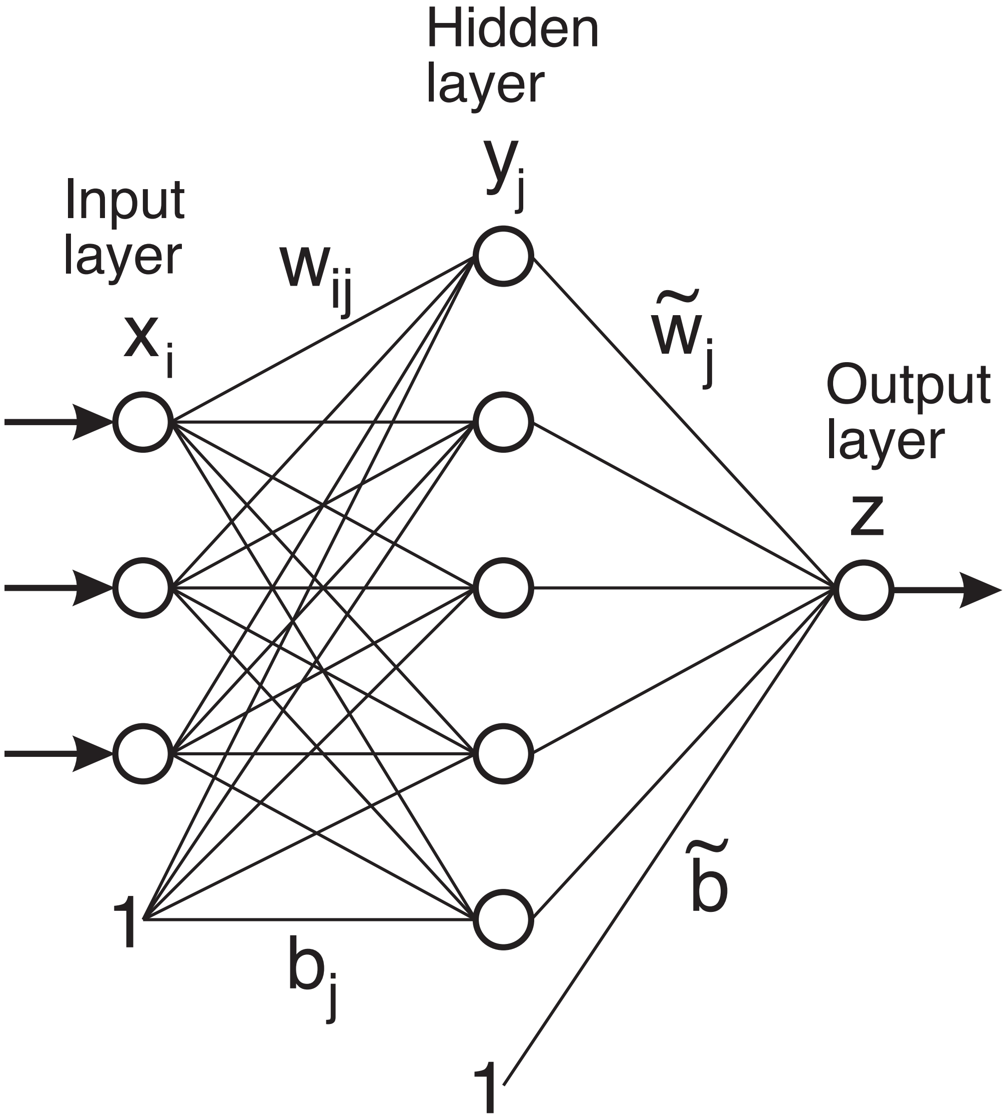


Figure 18: An example of a neural network model, where there are three neurons in the input layer, five in the hidden layer, and one in the output layer. The parameters w_{ij} and $\tilde{w}_j$ are the weights, and b_j and $\tilde{b}$ are the biases. The parameters b_j and $\tilde{b}$ can also be regarded as the weights for constant inputs of value 1.

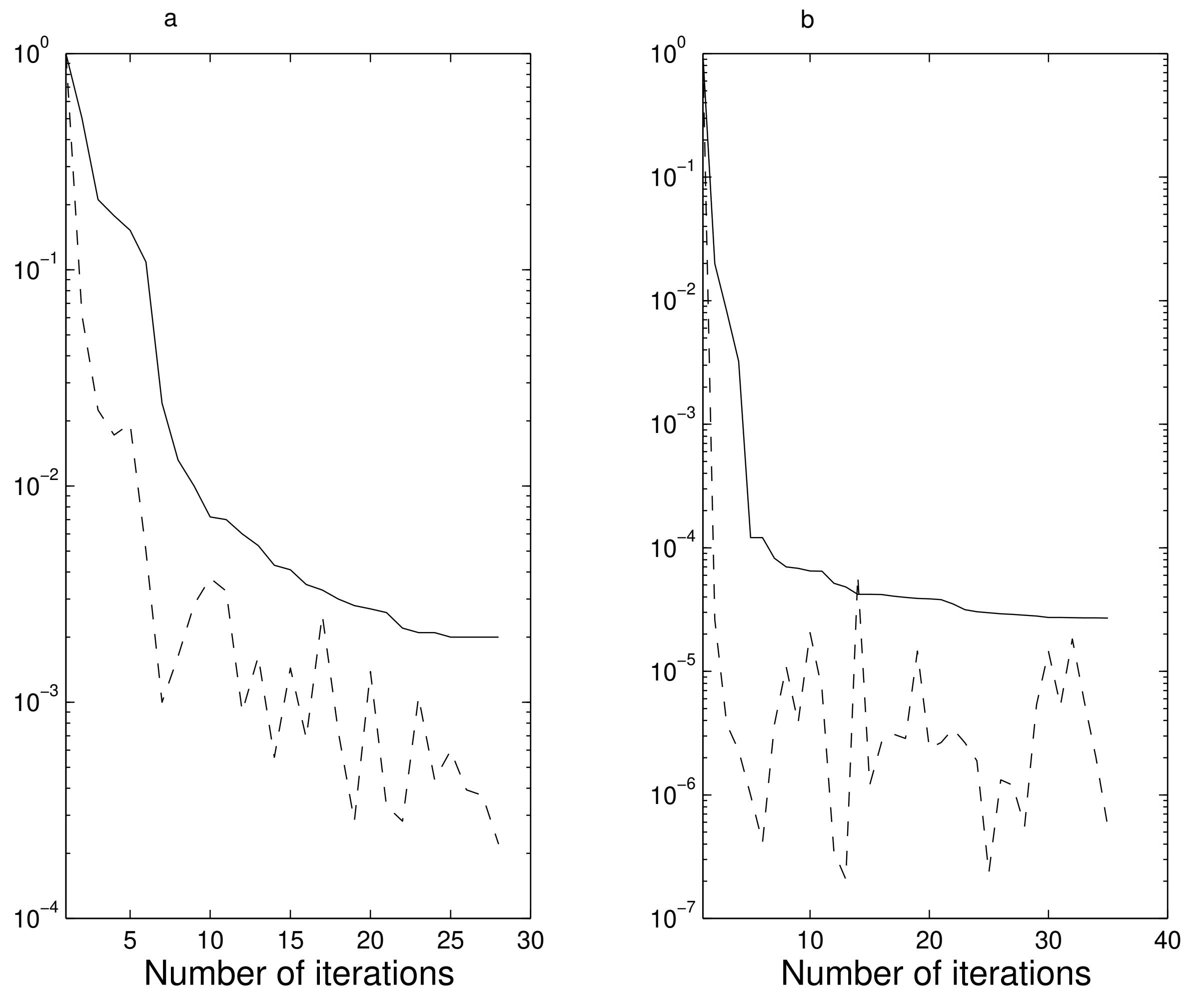


Figure 19: Variations of the normalized cost function (J/J_0) (solid line) and normalized gradient ($\|\mathbf{g}\|\|\mathbf{g_0}\|^{-1}$) (dash line) with the number of iterations for (a) the weakly nonlinear case of Fig. 7 and (b) the strongly nonlinear case of Fig. 11. The better initial guesses in (a) meant that the cost function and its gradients had relatively less distance to drop to convergence than in (b).